



**UNIVERSIDAD DE LAS CIENCIAS INFORMÁTICAS**

**FACULTAD 2**

*Desarrollo del módulo Configuración del Sistema de Gestión de Ensayos Clínicos*

**Trabajo de Diploma para optar por el título Ingeniero en Ciencias Informáticas**

**Autor:** Yudisley Garcia Guzmán

**Tutores:** Ing. Yisell Morera Rivera

Ing. Javier Villares Arias

**“La Habana, junio 2014”**

**“Año 56 de la Revolución”**



*“La conciencia es el resultado del conocimiento”*

*Hugo Rafael Chávez Frías*

## DECLARACIÓN DE AUTORÍA

*Declaro que soy la única autora de este trabajo y autorizo a la Universidad de las Ciencias Informáticas para que hagan el uso que estimen pertinente con este trabajo.*

*Para que así conste firmo la presente a los \_\_\_\_ días del mes de \_\_\_\_\_ del año \_\_\_\_\_.*

---

*Firma del autor*  
*Yudisley Garcia Guzmán*

---

*Firma del tutor*  
*Yisell Morera Rivera*

---

*Firma del tutor*  
*Javier Villares Arias*

## **SÍNTESIS DEL TUTOR**

### *Ing. Javier Villares Arias.*

*Graduado en el año 2009 en la Universidad de las Ciencias Informáticas de Ingeniero en Ciencias Informáticas, con categoría docente de Instructor. Durante su trayectoria laboral se ha desempeñado como profesor de las asignaturas Administración de Servidores Linux, Debates Históricos Contemporáneos y Práctica Profesional, también se desempeña en el Departamento de Sistemas Especializados en Salud (SES) en los roles de Desarrollador en el proyecto de Estomatología y Arquitecto del departamento. Ha prestado servicios en el Departamento de Gestión Hospitalaria en el desarrollo del sistema His.*

Correo Electrónico: [jvillares@uci.cu](mailto:jvillares@uci.cu)

### *Ing. Yisell Morera Rivera*

*Graduada de Ingeniería en Ciencias Informáticas en la Universidad de las Ciencias Informáticas (UCI) en el 2011. En la vinculación con la producción, como especialista, pertenece al Departamento de Sistemas Especializados en Salud (SES) del Centro de Informática Médica (CESIM) y específicamente trabaja en el proyecto Prótesis 2.0 donde se desempeña como desarrolladora.*

Correo electrónico: [ymorera@uci.cu](mailto:ymorera@uci.cu)

## **Agradecimientos**

*Quiero agradecer a mi mamá, por ser madre y amiga a la vez, por ayudarme y apoyarme en todas mis decisiones y acciones siempre que fueran correctas, por contribuir a construir la mujer que soy hoy día.*

*A mi papá Miguel, por educarme, darme cariño y enseñarme siempre. lo que es correcto*

*A mi hermana, por siempre ser tan especial y darme su cariño.*

*A mi hermano Cristian que lo adoro mucho y ser tan cariñoso conmigo.*

*A mis abuelos maternos, por estar ahí siempre que los necesito.*

*A mi abuela paterna que aunque no está hoy conmigo, la quiero mucho y la recuerdo siempre con tanto cariño por haber sido la mejor abuela del mundo.*

*A Yamila, la única amiga verdadera que he tenido, por siempre estar ahí conmigo en los momentos más crudos lejos de mi familia, por su cariño y ayuda incondicional.*

*A mi novio Arian, por la paciencia y la ayuda que me ha brindado siempre que la he necesitado. Por confiar en mí y darme las fuerzas para seguir hasta el final aunque en momentos las cosas se tornaran difíciles.*

*A Mayte, por ser la suegra y madre mas buena, admirable y especial del mundo.*

*A Humberto, por su cariño especial y darme los consejos necesarios para seguir adelante.*

*A Nardo y Arianna, por su cariño.*

*A mis amistades de la antigua aula en la que estaba por apoyarme y estar ahí siempre para mi.*

*A Kati, David, Claudia y Jorge Enrique, por su cariño y preocupación.*

*A mis tutores, por su ayuda incondicional y por darme los instrumentos esenciales para la construcción de este trabajo.*

*A Daniel, porque a pesar de que en ocasiones tuvimos algunas diferencias contribuyo a la realización de esta investigación, gracias por todo.*

*A todas aquellas personas que de una forma u otra me ayudaron y me dieron su ejemplo*

## **Dedicatoria**

*A mis padres que siempre quieren lo mejor para mí.*

*A mi hermanos, Cristian y Leydi, por su cariño.*

*A mis abuelos, por tanto cariño.*

*A mi abuela Luz Maria, por haber sido tan especial conmigo y por su cariño incondicional.*

*A mi novio, por tener tanta paciencia conmigo en mis momentos de estrés y por su apoyo incondicional*

*A mis suegros, por ser los mejores del mundo y por tanto cariño y apoyo.*

*A mi mejor amiga Yamila, por su cariño y apoyarme en todo momento.*

## **Resumen**

Hoy día en el Centro de Inmunología Molecular se encuentra desplegado el sistema Clínicas 1.0, el cual gestiona los ensayos clínicos llevados a cabo en esta institución. Dicho software cuenta con 4 módulos, entre los que se destaca Administrar Sistema. Este subsistema presenta deficiencias en su implementación como la definición estática de los roles y no existe la posibilidad de eliminar usuarios, centros y estudios; la tecnología usada es prácticamente obsoleta, y para su construcción no se utilizó un framework de desarrollo, por lo que su soporte sería sumamente costoso. Por todo lo anteriormente planteado, el CIM en colaboración con el Centro de Informática Médica (CESIM) radicado en la Universidad de las Ciencias Informáticas (UCI), deciden realizar una nueva versión titulada Sistema de Gestión de Ensayos Clínicos 1.0 (SIGEC).

Para el desarrollo de la solución propuesta, se utilizaron las herramientas y tecnologías definidas por el Departamento de Sistemas Especializados en Salud (SES), entre las cuales se encuentran: Java, como lenguaje de programación; Eclipse Ganymede, como entorno de desarrollo integrado; como metodología de desarrollo, Proceso Unificado de Desarrollo (RUP); como herramienta de modelado Visual Paradigm; para el análisis y diseño el Lenguaje Unificado de Modelado y como sistema de gestión de base de datos PostgreSQL.

El módulo Configuración permitirá garantizar con la inclusión de nuevas funcionalidades, la gestión de la información referente a los usuarios, roles, permisos, direcciones IP (Internet Protocol), centros y estudios y añadirá a SIGEC 1.0 atributos sólidos de seguridad superiores a los implementados en el sistema Clínicas 1.0.

## **Palabras Claves:**

Centro, Direcciones IP, Estudio, Permisos, Roles, Usuario.

# Tabla de contenidos

---

## TABLA DE CONTENIDOS

<b>INTRODUCCIÓN</b> .....	- 1 -
<b>CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA DE LOS PROCESOS DE DESARROLLO DE SOFTWARE DEL MÓDULO CONFIGURACIÓN DEL SISTEMA DE GESTIÓN DE ENSAYOS CLÍNICOS.</b> .....	- 7 -
1.1 Proceso de Autenticación, Autorización y Auditoría (AAA).....	- 7 -
1.2 Seguridad de aplicaciones web .....	- 8 -
1.3 Descripción del sistema Clínicas 1.0 .....	- 9 -
1.3.2 Módulo Administrar Sistema .....	- 10 -
1.4 Estudio de los sistemas informáticos para la gestión de la configuración en el ámbito nacional e internacional.....	- 10 -
1.5 Modelo de Madurez y Capacidades Integrado.....	- 13 -
1.6 Metodología de desarrollo de software.....	- 14 -
1.6.1 Metodología: Proceso Unificado de Desarrollo.....	- 14 -
1.6.2 Lenguaje Unificado de Modelado 2.0 .....	- 14 -
1.7 Herramientas utilizadas .....	- 15 -
1.8 Lenguajes de Programación.....	- 16 -
1.9 Tecnologías utilizadas para desarrollo del sistema.....	- 17 -
1.9.1 Frameworks .....	- 17 -
1.9.2 Librerías.....	- 19 -
1.10 Tecnologías horizontales.....	- 19 -
<b>CAPÍTULO 2: CARACTERÍSTICAS DEL MÓDULO CONFIGURACIÓN DEL SISTEMA DE GESTIÓN DE ENSAYOS CLÍNICOS.</b> .....	- 21 -
2.1 Propuesta de solución .....	- 21 -
2.2 Modelo de Dominio.....	- 22 -
2.2.1 Conceptos fundamentales del dominio .....	- 22 -
2.1.2 Diagrama del Modelo de Dominio.....	- 23 -
2.3 Especificación de los requisitos del sistema (ERS).....	- 23 -



# Tabla de contenidos

---

2.3.1 Requisitos funcionales del sistema.....	- 23 -
2.3.2 Requisitos no funcionales del sistema .....	- 25 -
2.4 Modelo de casos de uso del sistema.....	- 27 -
2.4.1 Definición de los actores del sistema .....	- 28 -
2.4.2 Diagrama de casos de uso del sistema .....	- 28 -
2.4.3 Descripción de los casos de uso.....	- 28 -
<b>CAPÍTULO 3: DISEÑO DEL MÓDULO CONFIGURACIÓN DEL SISTEMA DE GESTIÓN DE ENSAYOS CLÍNICOS</b> .....	- 35 -
3.1 Descripción de la arquitectura .....	- 35 -
3.1.1 Arquitectura cliente-servidor .....	- 35 -
3.1.2 Patrón arquitectónico: Modelo Vista Controlador (MVC).....	- 35 -
3.2 Modelo de diseño .....	- 36 -
3.2.1 Patrones de diseño.....	- 37 -
3.2.2 Diagrama de paquetes.....	- 38 -
3.2.3 Diagrama de clases del diseño.....	- 39 -
<b>CAPÍTULO 4: IMPLEMENTACIÓN DEL MÓDULO CONFIGURACIÓN DEL SISTEMA DE GESTIÓN DE ENSAYOS CLÍNICOS</b> .....	- 51 -
4.1 Modelo de Datos .....	- 51 -
4.1.1 Diagrama de Modelo de Datos.....	- 52 -
4.1.2 Descripción de las tablas de la base de datos .....	- 52 -
4.2 Modelo de Implementación.....	- 60 -
4.2.1 Diagrama de Despliegue .....	- 61 -
4.2.2 Diagrama de Componentes .....	- 61 -
4.3 Tratamiento de errores .....	- 63 -
4.4 Seguridad.....	- 64 -
4.5 Estrategias de codificación. Estándares y estilos a utilizar .....	- 64 -
<b>CONCLUSIONES</b> .....	- 67 -
<b>RECOMENDACIONES</b> .....	- 68 -
<b>REFERENCIAS BIBLIOGRÁFICAS</b> .....	- 69 -

## *Tabla de contenidos*

---

<b>BIBLIOGRAFÍA</b> .....	- 74 -
<b>ANEXO 1</b> .....	- 80 -
<b>ANEXO 2</b> .....	- 87 -

# *Introducción*

---

## **INTRODUCCIÓN**

A lo largo de la historia siempre ha existido la necesidad de innovar y perfeccionar mecanismos para la comunicación, haciéndolos más eficientes y eficaces. Con el paso del tiempo se han desarrollado las Tecnologías de la Información y las Comunicaciones (TIC), las cuales se pueden definir de forma operativa como un conjunto de técnicas, desarrollos y dispositivos avanzados que integran funcionalidades de almacenamiento, procesamiento y transmisión de datos (1). Las TIC en la actualidad están provocando profundos cambios y transformaciones de naturaleza social y cultural, además de económicos, incidiendo directamente sobre los principales sectores de la sociedad.

Hoy día en el sector de la salud, con los grandes avances tecnológicos surgidos se han obtenido mejores servicios que antes eran imposibles o inalcanzables. La integración de las TIC en esta área se ha convertido en un tema interesante no solo para médicos o instituciones sino para la sociedad en general, logrando revolucionar la manera tradicional del servicio que ofrecían dichas instituciones sobre sus pacientes, permitiendo la construcción de una nueva forma de organización y funcionamiento de los mismos, donde la innovación sea factor clave para un cambio que permita transformar la administración tradicional en la salud mediante procesos más eficientes orientados a la atención del ciudadano y al logro de un sistema mejor comunicado y más inteligente. (1)

Existen a nivel mundial numerosos centros de investigaciones científicas, los cuales tributan al campo de la salud, en los que se llevan a cabo una serie de estudios sobre el comportamiento de diversas enfermedades con el objetivo de crear fármacos que puedan contrarrestarlas, así como la prevención y tratamiento de estas. Para la creación de dichos fármacos se realizan los llamados Ensayos Clínicos (EC), los cuales en esencia, son estudios en los que se comparan mediante la medición de varios parámetros, los efectos de dos o más tratamientos o intervenciones terapéuticas en un grupo homogéneo de personas con una enfermedad o situación médica similar, con el objetivo de detectar si uno de los tratamientos comparados es más o menos beneficioso que el otro.

Los ensayos clínicos surgen como la respuesta científica a la necesidad ética de garantizar la eficacia y la seguridad de los tratamientos que reciben los pacientes. Ellos proporcionan un método controlado, objetivo y reproducible para medir los efectos de un tratamiento sobre la salud y así permiten establecer con objetividad una opinión sobre su utilidad en situaciones de enfermedad o prevención, además requieren un cuidado extremo, porque su condición de experimentación le confiere una trascendencia

## *Introducción*

---

ética por implicar a seres humanos y por la enorme repercusión que puede derivarse de sus resultados, al poder éstos condicionar el tipo de tratamiento a recibir por miles de pacientes. (2)

Aunque la mayoría de estos centros se encuentran en países con un alto nivel de desarrollo, Cuba a pesar del riguroso bloqueo al que ha estado sometida a lo largo de los años de Revolución, también ha incursionado y ha obtenido grandes logros en este campo a través de la creación de varios centros especializados, entre los que se encuentra el Centro de Inmunología Molecular (CIM) inaugurado el 5 de Diciembre de 1994, con el objetivo de llevar a cabo la producción de nuevos biofármacos para el tratamiento del cáncer y otras enfermedades crónicas no transmisibles, con el fin de introducirlos en la salud pública cubana y realizar aportes importantes a su economía, haciendo de la actividad científica una rama productiva económicamente sostenible.

Este centro lleva la conducción de varios ensayos clínicos, en los que se maneja una cantidad importante de información la cual debe ser recogida durante el estudio de cada ensayo realizado, y los documentos generados se almacenan durante los 15 años posteriores a la realización del estudio. Con el objetivo de llevar un minucioso control de tal volumen de documentación, los médicos del CIM en conjunto con el Centro de Informática Médica (CESIM) radicado en la Universidad de las Ciencias Informáticas (UCI) y encargado de desarrollar aplicaciones para el sector de la Salud, deciden trabajar unidos en la informatización de este proceso.

En el 2012 se desplegó la herramienta elaborada por el centro CESIM, capaz de gestionar la información almacenada y erradicar en gran medida los problemas que se puedan presentar, la misma lleva como nombre Clínicas en su versión 1.0.

El sistema Clínicas 1.0 cuenta con un conjunto de funcionalidades entre las cuales se encuentran: la gestión de centros, estudios, usuarios, direcciones IP (Internet Protocol, por sus siglas en inglés), además de la gestión de las Hojas CRD (Cuaderno de Recogida de Datos). Todas estas funcionalidades incluyendo otras pero no menos importantes se encuentran distribuidas por diferentes módulos: Administrar Sistema, Gestionar Estudio, Enviar Datos y Extraer Datos.

El Módulo Administrar Sistema, es el encargado de la configuración del software, el mismo permite gestionar los permisos y roles de los usuarios que accederán al sistema, así como la gestión de direcciones o rangos de IP para los usuarios, centros y estudios creados.

Los especialistas al interactuar con el módulo, detectan inconvenientes en la forma en la que se gestionan los usuarios, estudios y centros, pues estas funcionalidades no admiten la eliminación de entidades, solo

## *Introducción*

---

que se desactiven, lo que trae consigo que se tenga información innecesaria en el sistema y que le muestre al usuario que está trabajando con la aplicación información prescindible la cual puede crear confusión. Por otra parte, los roles se definen estáticos, acarreado como consecuencia que cada vez que se desee crear un nuevo rol haya que reprogramar esta parte del software, siendo imposible asignarle o quitarle responsabilidades a los roles existentes y si el usuario que trabaja con la aplicación requiere de otro rol, no puede interactuar con la misma, afectando significativamente la adaptabilidad del software a las necesidades del cliente.

Teniendo en cuenta estas deficiencias y que en la confección de este sistema no se utilizó un framework de desarrollo, además el servidor de aplicaciones empleado para su realización Apache Tomcat 5.5 es obsoleto en comparación con la versión actual 7.0, por lo que el soporte de dicha aplicación sería sumamente costoso. Con el objetivo de suprimir estos inconvenientes el CESIM decide desarrollar una nueva versión, la cual lleva el nombre de Sistema de Gestión de Ensayos Clínicos 1.0 (SIGEC).

Lo anteriormente planteado define como **problema a resolver**: ¿Cómo lograr la configuración de roles, usuarios, estudios, centros, permisos y direcciones IP (Internet Protocol) para garantizar el proceso de autenticación y autorización en el Sistema de Gestión de Ensayos Clínicos?

Centrado en el **objeto de estudio**: proceso de Autenticación, Autorización y Auditoría.

El **campo de acción** se enfoca en la gestión de la configuración de roles, usuarios, estudios, permisos, centros y direcciones IP.

Para dar solución al problema se plantea como **objetivo general**: desarrollar el módulo Configuración del Sistema de Gestión de Ensayos Clínicos permitiendo la gestión de roles, usuarios, estudios, permisos, centros y direcciones IP, para garantizar el proceso de autenticación y autorización en dicho software.

Para dar cumplimiento al objetivo planteado se proponen las siguientes tareas de la investigación:

- Analizar el estado del arte de los sistemas de configuración y administración a nivel internacional y nacional, estableciendo similitudes con la investigación en curso.
- Asimilar la metodología, las herramientas y tecnologías, definidas por el Departamento de Sistemas Especializados en Salud (SES).

# Introducción

---

- Elaborar los artefactos correspondientes a los flujos de trabajo: Modelado de Negocio, Requerimientos, Análisis y Diseño e Implementación propuestos por la metodología de desarrollo Proceso Unificado de Desarrollo de Software (RUP).
- Implementar las funcionalidades correspondientes a la gestión de estudios y centros pertenecientes al módulo Configuración, aplicando las pautas de diseño e implementación definidas por el Departamento de Sistemas Especializados en Salud y siguiendo lo establecido en la especificación de requisitos.

En el desarrollo de la presente investigación se han empleado los siguientes métodos científicos:

## Métodos teóricos

- Analítico – sintético: tiene como objetivo analizar las teorías y documentos, permitiendo la extracción de los elementos más importantes que se relacionan con el proceso de Autenticación, Autorización y Auditoría en aplicaciones web. El mismo fue empleado en la realización de toda la tesis.
- Inductivo – deductivo: son formas de razonamiento que permiten llegar a un grupo de conocimientos generalizadores, tanto desde el análisis de lo particular a lo general, como desde el análisis de elementos generalizadores a uno de menor nivel de generalización. Dicho método fue puesto en práctica en toda la investigación.
- Análisis histórico- lógico: estudia la trayectoria histórica real de los fenómenos, su evolución y desarrollo. Dicho método se ve reflejado en el capítulo 1 de la tesis, en el epígrafe 1.3 Síntesis de los sistemas informáticos existentes a nivel nacional e internacional.
- Modelación: permite la creación de modelos (propuestas, alternativas, estrategias), es una reproducción simplificada y sintética de la realidad. El mismo fue empleado en los capítulos 2, 3, 4 de la presente investigación.

## Métodos empíricos

- Observación: registro visual de lo que ocurre en una situación real, en un fenómeno determinado, clasificando y consignando los hechos y acontecimientos pertinentes de acuerdo con algún esquema previsto. El mismo fue puesto en práctica en toda la investigación.
- Entrevista: es una conversación planificada para obtener información. Su uso constituye un medio para el conocimiento cualitativo de los fenómenos o sobre características personales del entrevistado. Este

# *Introducción*

---

se utilizó para obtener información del negocio, a través de interrogantes realizadas al jefe de proyecto Erilán Martínez Jera, ver [Anexo 1](#).

Con la introducción de la solución propuesta en la gestión de la información de los ensayos clínicos se obtendrán los siguientes beneficios:

- Permitirá la gestión dinámica de la información referente a los usuarios, estudios, centros, roles y direcciones IP (Internet Protocol).
- Aumentará la calidad de la gestión de la información de los ensayos clínicos, posibilitando a los especialistas del Centro de Inmunología Molecular la creación de nuevos roles sin necesidad de reprogramar esta parte del sistema.
- Facilitará a los usuarios la eliminación de entidades, evitando la visualización de información prescindible, y la confusión que la misma pueda proporcionarle.
- Generará un ahorro significativo de capital monetario, posibilitando la adquisición de un software de desarrollo propio, garantizando a su vez soporte o mantenimiento sin incurrir en grandes gastos por concepto de licencias o servicios.
- Contribuirá a un desarrollo más certero de los ensayos, con una mayor validez en la investigación, lo que permitirá estimar con mayor veracidad el impacto de los mismos en la salud de los pacientes.

El presente documento está estructurado por capítulos de la siguiente forma:

**CAPÍTULO 1. FUNDAMENTACIÓN TEÓRICA DE LOS PROCESOS DE DESARROLLO DE SOFTWARE DEL MÓDULO CONFIGURACIÓN DEL SISTEMA DE GESTIÓN DE ENSAYOS CLÍNICOS:** se realiza una breve descripción de los elementos que conforman el proceso de Autenticación, Autorización y Auditoría, así como también una descripción del sistema Clínicas 1.0 que se encuentra desplegado en el CIM. Se desarrolla un estudio del estado del arte de los sistemas web, tanto a nivel internacional como nacional, que tengan un vínculo directo con el trabajo realizado en la presente investigación y se analizan las tecnologías y metodología de software que serán utilizadas para dar solución al problema planteado.

**CAPÍTULO 2. CARACTERÍSTICAS DEL MÓDULO CONFIGURACIÓN DEL SISTEMA DE GESTIÓN DE ENSAYOS CLÍNICOS:** se realiza una presentación visual de las clases conceptuales a través del modelo

## *Introducción*

---

de dominio, se presenta la especificación de los requisitos funcionales y no funcionales del sistema, se definen además los casos de uso para así conocer las funcionalidades con las que contará el software.

### **CAPÍTULO 3. DISEÑO DEL MÓDULO CONFIGURACIÓN DEL SISTEMA DE GESTIÓN DE ENSAYOS**

**CLÍNICOS:** se describen y se analizan detalladamente los aspectos relacionados con la estructura de la solución propuesta para dar respuesta a la problemática en cuestión. Se fundamenta la arquitectura empleada, además de una descripción de la gestión de usuarios, centros, estudios, roles, permisos y direcciones IP a través de varios artefactos como el diagrama de paquetes y el diagrama de clases del diseño, se brinda además una descripción de las principales clases del diseño empleadas en el desarrollo del módulo Configuración.

### **CAPÍTULO 4. IMPLEMENTACIÓN DEL MÓDULO CONFIGURACIÓN DEL SISTEMA DE GESTIÓN DE**

**ENSAYOS CLÍNICOS:** se expone la implementación del sistema propuesto a raíz de los resultados obtenidos en el diseño. Se elabora el modelo de datos y se describen las tablas que lo integran, además se obtienen los diagramas de componentes y despliegue, así como las estrategias de codificación, estándares y el estilo de código a utilizar.



# *Fundamentación teórica de los procesos de desarrollo de software del módulo Configuración del Sistema de Gestión de Ensayos Clínicos*

---

## **CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA DE LOS PROCESOS DE DESARROLLO DE SOFTWARE DEL MÓDULO CONFIGURACIÓN DEL SISTEMA DE GESTIÓN DE ENSAYOS CLÍNICOS.**

En el presente capítulo se realiza una breve descripción de los elementos que conforman el proceso de Autenticación, Autorización y Auditoría, así como también una descripción del sistema Clínicas 1.0 que se encuentra desplegado en el CIM. Se desarrollará el estudio del estado del arte de los sistemas web, tanto a nivel internacional como nacional que tengan un vínculo directo con el trabajo realizado en la presente investigación. Se describe además el análisis de las tecnologías y metodología de software que serán utilizadas para dar solución al problema en cuestión.

### **1.1 Proceso de Autenticación, Autorización y Auditoría (AAA)**

Este proceso se definió como un marco de funcionamiento común para gestionar las operaciones de Autenticación, Autorización y Auditoría en un escenario de control de acceso a la red, permite el acceso de los usuarios legítimos a los activos conectados a la red e impide el acceso no autorizado.(3)

**Autenticación:** cuando un usuario o cliente requiere acceso a la red, debe presentar información personal que permita establecer de manera unívoca su identidad dentro del entorno. El usuario podrá presentar cualquier tipo de credencial de identidad (nombre de usuario y contraseña, certificado de identidad secreto o compartido), usando para ello algunos de los mecanismos de control de acceso. El servidor AAA y concretamente su servicio de autenticación, deberá obtener dichas credenciales y de forma interna (mediante bases de datos, políticas de autenticación o servicios auxiliares), determinar si ese usuario es realmente quien dice ser. (4)

**Autorización:** el proceso de autorización deberá determinar si el usuario, una vez autenticado (si es necesario), tiene o no derecho a hacer uso del servicio solicitado. Para ello, la decisión de autorización puede tomarse en función de diferentes credenciales del usuario. Por ejemplo, la autorización puede otorgarse simplemente comprobando que el usuario ha sido autenticado correctamente, aunque en entornos más especializados puede otorgarse en base a credenciales de autorización, como pueden ser sentencias de atributos que identifican el rol o papel que juega el usuario dentro de la organización. En este último caso, la manera en que el servicio AAA recupera esta información de autorización, depende del escenario concreto de acceso, ya que puede ser presentada directamente por el usuario o recuperada desde repositorios bien conocidos. (4)

# *Fundamentación teórica de los procesos de desarrollo de software del módulo Configuración del Sistema de Gestión de Ensayos Clínicos*

---

**Auditoría:** existen entornos donde se requiere llevar un control detallado de las operaciones que realizan los usuarios de los sistemas, por ejemplo, en un entorno de alta seguridad donde toda la información operacional debe ser almacenada. En este tipo de entornos, cualquier información relativa a una sesión de usuario (acceso al servicio, tipo de servicio obtenido, cantidad de tráfico enviado o recibido, tiempo de conexión), debe almacenarse para que después pueda ser usada apropiadamente durante el proceso de auditoría. (4)

La presente investigación se centrará en la Autenticación y la Autorización, pues la Auditoría, el sistema SIGEC 1.0 lo tiene implementado en el módulo Extracción.

## **1.2 Seguridad de aplicaciones web**

Hoy día la Web se ha convertido en un importante medio de comunicación e intercambio de información, que ofrece un acceso abierto a un conjunto de datos que explícitamente se hacen públicos. Para estos sistemas la principal vulnerabilidad que se presenta es que la información viaja a través de la red, lo que implica que puede ser sustraída por cualquier persona, no obstante, existe una serie de vulnerabilidades como la falsificación de identidad, el robo de sesiones autenticadas, la falsificación o modificación de los mensajes. Por otra parte otro aspecto que cobra especial importancia es la seguridad de la información que se intercambia en la Web, que cada vez está más expuesta, a personas que incurrir en delitos informáticos. (4)

Los principios básicos de seguridad que debe cumplir cualquier aplicación o servicio web son:

**Validación de la entrada y salida de información:** la entrada y salida de información es el principal mecanismo que dispone un atacante para enviar código malicioso contra el sistema. Por tanto, siempre debe verificarse que cualquier dato entrante o saliente es apropiado y que el mismo esté en el formato que se espera. Las características de estos datos deben estar predefinidas y deben verificarse en todas las ocasiones.

**Diseños simples:** los mecanismos de seguridad deben diseñarse para que sean lo más sencillos posibles, evitando elementos que compliquen excesivamente el trabajo a los usuarios. Si los pasos necesarios para proteger de forma adecuada una función o módulo son muy complejos, la probabilidad de que estos no se ejecuten de la forma esperada será muy elevada.

# *Fundamentación teórica de los procesos de desarrollo de software del módulo Configuración del Sistema de Gestión de Ensayos Clínicos*

---

**Utilización y reutilización de componentes de confianza:** debe potenciarse el principio de la reutilización en todos los casos en que sea posible. Por tanto, cuando exista un componente que resuelva un problema determinado correctamente, lo más adecuado es reutilizarlo.

**Defensa en profundidad:** nunca confiar en que un componente realizará su función de forma permanente y ante cualquier situación. Es necesario disponer de los mecanismos de seguridad suficientes para que, cuando un componente del sistema falle ante un determinado evento, otros sean capaces de detectarlo.

**Proveer servicios de autenticación segura a las aplicaciones Web:** las aplicaciones deberán poseer reglas o mecanismos de autenticación donde los certificados de seguridad establecidos puedan poseer variables que se modifiquen durante el tiempo en que la sesión se encuentre activa.

**Verificación de privilegios:** los sistemas deben diseñarse para que funcionen con la menor cantidad de privilegios posibles. Igualmente, es importante que los procesos únicamente dispongan de los privilegios necesarios para desarrollar su función, de forma que queden compartimentados.

**Ofrecer la mínima información:** ante una situación de error o una validación negativa, los mecanismos de seguridad deben diseñarse para que faciliten la mínima información posible. De la misma forma, estos mecanismos deben estar diseñados para que una vez denegada una operación, cualquier acción posterior sea igualmente denegada. (4)

## **1.3 Descripción del sistema Clínicas 1.0**

A continuación se profundizará en los aspectos del sistema Clínicas 1.0 y los módulos que lo conforman. Este sistema surge, producto a un acuerdo de los especialistas del CIM en conjunto con el Centro de Soluciones de Informática Médica (CESIM), encargado de desarrollar aplicaciones para el Ministerio de Salud Pública en Cuba, radicado en la Universidad de las Ciencias Informáticas (UCI). El software se desarrolló con el objetivo de informatizar todo el proceso de la gestión de la información de los ensayos clínicos.

Este producto hoy día se encuentra desplegado en el Centro de Inmunología Molecular, el cual ayuda a los especialistas de esta institución a la gestión de la información de los ensayos clínicos. El mismo cuenta con cuatro módulos: Administrar Sistema, Gestionar Estudio, Enviar Datos y Extraer Datos, agrupando un conjunto de funcionalidades entre las cuales se encuentran: la gestión de usuarios, estudios, centros, direcciones IP (Internet Protocol, por sus siglas en inglés), además de la gestión de las Hojas CRD (Cuaderno de Recogida de Datos).

# *Fundamentación teórica de los procesos de desarrollo de software del módulo Configuración del Sistema de Gestión de Ensayos Clínicos*

---

Cada uno de ellos ocupa un papel fundamental en software Clínicas 1.0 como por ejemplo: Administrar Sistema, es el encargado de la gestión de los usuarios, direcciones IP y roles que intervienen en el sistema; Gestionar Estudio maneja la modificación de datos, restauración, visualización de todos los estudios especificados así como la gestión de cronogramas de ejecución del ensayo. Por otra parte el módulo Enviar Datos realiza la gestión de las Hojas CRD, de los sujetos que intervienen en el ensayo clínico, y la generación de notas (generales, de monitoreo), por último Extraer Datos gestiona los conjuntos de datos creados.

## **1.3.2 Módulo Administrar Sistema**

Este subsistema tiene como objetivo procesar de forma segura la configuración del software Clínicas 1.0 y de gestionar los permisos y roles de los usuarios que accederán al sistema, maneja las direcciones o rangos de IP asignadas a los usuarios, centros y estudios creados, también se encarga de enviar notificaciones por correo a los usuarios, con una descripción de los datos necesarios para acceder al sistema.

Este módulo carece de funcionalidades que permitan realizar una adecuada gestión de usuarios, estudios y centros, teniendo en cuenta que permite solamente la desactivación de estas entidades, por lo que no es posible la eliminación de las mismas, trayendo como resultado que se tenga información innecesaria en la aplicación. Por otra parte, los roles se definen estáticos, acarreado como consecuencia que cada vez que se desee crear un nuevo rol haya que reprogramar esta parte del sistema, siendo imposible asignarle o quitarle responsabilidades a los roles existentes y si el usuario que trabaja con la aplicación requiere de otro rol, no puede interactuar con la misma, afectando significativamente la adaptabilidad del software a las necesidades del cliente.

## **1.4 Estudio de los sistemas informáticos para la gestión de la configuración en el ámbito nacional e internacional.**

A nivel nacional e internacional han sido diseñadas y confeccionadas varias soluciones informáticas, con el fin de fortalecer los procesos de Autenticación Autorización y Auditoría. A continuación se realiza una breve reseña de algunos de estos sistemas.

# *Fundamentación teórica de los procesos de desarrollo de software del módulo Configuración del Sistema de Gestión de Ensayos Clínicos*

---

## **Ámbito Internacional**

### **OpenClinica**

Es un software multiplataforma de captura electrónica de datos (EDC, por sus siglas en inglés) y de gestión de datos para ensayos clínicos y otros estudios de investigación. Este sistema lleva la configuración de las copias de seguridad automáticas, de los distintos centros y de las diferentes visitas del estudio, así como de la plataforma según las necesidades particulares del estudio y de los usuarios, para que cada perfil de usuario pueda solamente realizar las tareas que debe realizar. De esta manera éstos (promotores, coordinadores, investigadores, monitores y gerente de datos) tendrán restringidas algunas funcionalidades de lectura o escritura en la base de datos o el acceso a determinadas funciones de la aplicación. Puede ser configurado para enviar automáticamente mensajes de correo electrónico. Por ejemplo, si un usuario pierde su contraseña, la contraseña se puede restablecer y un correo puede ser enviado para notificar al usuario (5). Para el desarrollo de este sistema informático se utilizó como lenguaje de programación Java, utilizando Eclipse 3.4 como entorno de desarrollo, el servidor web es Apache Tomcat 5.5, herramienta para el modelado de procesos Visual Paradigm 6.4 y la base de datos esta creada en PostgreSQL 8.2.

Esta aplicación a pesar de sus potencialidades no se adapta a la forma en la que se realizan los ensayos clínicos en el CIM, además las herramientas que fueron utilizadas para su confección no coinciden con las definidas para el desarrollo del módulo.

## **Ámbito Nacional**

### **Clínicas**

Este sistema permite acelerar y automatizar la conducción de ensayos clínicos en el CIM. El mismo cuenta con cuatro módulos entre los que se destaca Administrar sistema, el cual se encarga de la gestión de usuarios, centro, estudios y permite gestionar los permisos y roles de los usuarios que accederán a la aplicación, así como la administración de direcciones IP desde las cuales se accederá al software. Clínicas 1.0 fue desarrollado en el 2012 en la UCI y para su desarrollo se utilizó como lenguaje de programación Java; Apache Tomcat 5.5, como servidor web; y como gestor de base de datos PostgreSQL 8.2. (6)

# *Fundamentación teórica de los procesos de desarrollo de software del módulo Configuración del Sistema de Gestión de Ensayos Clínicos*

---

A partir de las experiencias obtenidas por especialistas mediante la interacción con este sistema, se decide modificar las funcionalidades del mismo adaptándolas a las nuevas necesidades del cliente, para conformar la nueva versión del módulo Configuración del sistema SIGEC 1.0.

## **HIS**

El Sistema de Información Hospitalaria (HIS, por sus siglas en inglés) desarrollado en CESIM, ubicado en la UCI permite la recolección, almacenamiento, procesamiento, recuperación y comunicación de información de atención al paciente para todas las actividades relacionadas con la institución de salud. El mismo cuenta con una serie de módulos, entre los cuales se encuentra Configuración, que permite la administración del sistema, posibilitando configurar las funcionalidades de cada uno de los módulos. Garantiza la seguridad de acceso al sistema, posibilitando a su vez la administración de usuarios y roles, así como la estructura organizativa y de servicios del hospital. Además gestiona los datos generales de la institución hospitalaria y visualiza la bitácora del sistema. (7)

Para la construcción del software se utilizó Java como lenguaje de programación, así como una serie de herramientas entre las cuales se encuentran: PostgreSQL 8.4, como servidor de base de datos, Eclipse 3.4.2 Ganymede como entorno de desarrollo, Visual Paradigm 6.4 para el modelado de los artefactos.

Una vez analizados los requisitos funcionales que posee el módulo Configuración del HIS, se decide modificar las funcionalidades: Adicionar usuarios, Buscar usuario, Eliminar usuario, Modificar usuario, Ver detalles de usuario, Adicionar rol, Buscar rol, Eliminar rol, Modificar rol, Ver detalles de rol. Por otra parte el código de las funcionalidades antes mencionadas será adaptado a los requisitos específicos del módulo Configuración del sistema SIGEC 1.0 y la acción Configurar funcionalidades será reutilizada pero no sufrirá modificación alguna.

Al concluir el estudio de estos sistemas se pudo comprobar que los requisitos funcionales que contienen no son factibles para dar solución a la problemática identificada, pues no se adaptan a las necesidades del cliente, además las herramientas que fueron utilizadas para su confección en su mayoría son propietarias y no coinciden con las definidas para el desarrollo del módulo. A pesar de ser de código abierto, multiplataforma, no se adaptan a la forma en que llevan a cabo el proceso de ensayos clínicos en el CIM. Con el objetivo de aprovechar las utilidades y experiencias obtenidas por los especialistas con el despliegue del Clínicas 1.0 en el CIM, el equipo de trabajo determinó modificar las funcionalidades del

# *Fundamentación teórica de los procesos de desarrollo de software del módulo Configuración del Sistema de Gestión de Ensayos Clínicos*

---

mismo adaptándolas a las nuevas necesidades del cliente, para conformar la nueva versión del módulo Configuración del sistema SIGEC 1.0.

Después de analizar características y ventajas que posee el HIS, se decidió desarrollar el módulo Configuración del producto SIGEC 1.0 reutilizando y modificando funcionalidades del módulo Configuración del HIS, las cuales permitirán disminuir el tiempo de desarrollo y posibilitará una futura integración entre ambos sistemas.

## **1.5 Modelo de Madurez y Capacidades Integrado**

El Modelo de Madurez y Capacidades Integrado (CMMI, por sus siglas en inglés) es un enfoque de mejora de procesos que provee a las organizaciones de los elementos esenciales para un proceso efectivo, el cual mide la madurez del desarrollo del software con 5 niveles escalonados, cada área de proceso se asocia a uno de estos niveles y a medida que la institución cumple con los procesos definidos para cada uno, alcanza el nivel de madurez de referencia. Una organización alcanza un nivel determinado cuando ha puesto en práctica todas y cada una de las áreas de proceso aplicables a ese nivel y a los niveles inferiores. Este modelo integra disciplinas como sistemas y software en un solo marco de trabajo y describe además formas efectivas y probadas de hacer las cosas, no es un enfoque radical. Con la correcta utilización de este proceso de mejora se pueden alcanzar grandes ventajas como producir servicios y productos de alta calidad, mejorar la satisfacción del cliente e incrementar la participación en el mercado. (8)

La UCI certificó al nivel 2 de CMMI y con la ejecución del mismo se trabajó en algunos de los procesos que propone el nivel 3. Este programa de mejora que se despliega actualmente en la UCI tiene como objetivo la definición e implementación de los procesos necesarios para así reducir varios de los problemas que están vigentes en la producción de software en la Universidad. Dicho programa brinda una serie de políticas a cumplir por cada área de proceso. Por ello se creó un documento en la UCI llamado "0523\_Políticas", donde se mencionan las mismas, así como las que complementan la realización de estas áreas de proceso, dentro de las cuales se encuentra "10.6 Definir configuración de la metodología a utilizar" donde el proyecto debe definir y documentar la configuración de la metodología a utilizar: fases, actividades, roles y artefactos, para dar cumplimiento a estas, el Departamento de Sistemas Especializados en Salud definió que la metodología a utilizar sería RUP.



# *Fundamentación teórica de los procesos de desarrollo de software del módulo Configuración del Sistema de Gestión de Ensayos Clínicos*

---

## **1.6 Metodología de desarrollo de software.**

Para el desarrollo de aplicaciones informáticas se han elaborado varios estándares que marcan las pautas a seguir para un correcto modelado e implementación de los sistemas. Para poder realizar este proceso correctamente se deben tener en cuenta las características del software a desarrollar a la hora de elegir por cuáles de todos los estándares existentes se puede regir el proceso en cuestión. Por ello en este epígrafe se expondrá una síntesis de la metodología definida para el desarrollo del sistema Clínicas 1.0.

### **1.6.1 Metodología: Proceso Unificado de Desarrollo.**

Como metodología de desarrollo de software se utiliza el Proceso Unificado de Desarrollo (RUP, por sus siglas en inglés), es la más apropiada para el desarrollo de este sistema, pues está diseñada para la construcción de proyectos grandes.

RUP está basado en componentes e interfaces bien definidas, y apoyándose en el Lenguaje Unificado de Modelado (UML, por sus siglas en inglés), constituye una de las metodologías estándares más populares para el desarrollo de sistemas orientados a objetos. Es un marco del proyecto que describe una clase de los procesos que son iterativos e incrementales. Define una cantidad entera de las actividades y de los artefactos que se necesitan para conformar el proceso individual. La ventaja principal de RUP es que se basa en las mejores prácticas que se han intentado y se han probado en el campo de la informática.

El proceso de desarrollo especificado en RUP está dividido en cuatro fases: inicio, se define el alcance del proyecto; elaboración, se analiza y diseña la solución; construcción, se implementa la solución y transición, donde se concluye el proyecto y se lleva a la producción.

Cuenta con nueve flujos de trabajo, de ellos seis de ingeniería (Ingeniería de Negocios, Requerimientos, Análisis y Diseño, Implementación, Pruebas e Instalación) y tres de apoyo (Configuración y administración del cambio, Administración de proyecto y Ambiente), distribuidos en las cuatro fases de desarrollo antes mencionadas. (9)

### **1.6.2 Lenguaje Unificado de Modelado 2.0**

El Lenguaje Unificado de Modelado (UML, por sus siglas en inglés) es utilizado para visualizar, especificar y documentar cada una de las partes que comprende el desarrollo de software. Este no es una guía para realizar el análisis y diseño orientado a objetos, es decir, no es un proceso, es un lenguaje que permite la modelación de sistemas con tecnología orientada a objetos. (10)



# *Fundamentación teórica de los procesos de desarrollo de software del módulo Configuración del Sistema de Gestión de Ensayos Clínicos*

---

El mismo intenta solucionar el problema de propiedad de código que ocurre con muchos desarrolladores. Al implementar un lenguaje de modelado común para todos los desarrollos se crea una documentación común de igual manera, que cualquier desarrollador con conocimientos de UML será capaz de entender, independientemente del lenguaje utilizado para este.

Entre las características principales que presenta UML se encuentran: es unificado para la modelación de sistemas y permite la corrección de errores viables en todas las etapas. (11)

## **1.7 Herramientas utilizadas**

Para el desarrollo de este sistema se utilizan una serie de herramientas las cuales están descritas a continuación.

### **Eclipse 3.4.2 Ganymede**

Es un entorno de desarrollo portable, de código abierto y multiplataforma. Diseñado por la empresa IBM (International Business Machines) y actualmente sujeto a la Fundación Eclipse. Posee nuevas funcionalidades en dependencia de las necesidades o lo que se desee obtener. Permite además la vinculación con la herramienta de modelado Visual Paradigm mediante la Edición Empresarial, lo que proporciona un mejor entendimiento entre analistas y desarrolladores.

Entre sus principales ventajas se destacan que el Eclipse emplea módulos (en inglés plugin) para proporcionar toda su funcionalidad al frente de la plataforma de cliente rico, a diferencia de otros entornos monolíticos donde las funcionalidades están todas incluidas, las necesite el usuario o no. Este mecanismo de módulos constituye una plataforma ligera para componentes de software. (12)

### **JBoss Tools**

Son un conjunto de extensiones para eclipse que ofrecen soporte para desarrollar, desplegar y probar aplicaciones basadas en proyectos JBoss.org. Es un producto que provee un soporte completo para desarrollar aplicaciones de Java SE (Java Standard Edition) y Java EE (Java Enterprise Edition), además de sostener el desarrollo de cada capa de una aplicación empresarial/escritorio brinda componentes como editores, asistentes de instalación, marcado de sintaxis, tareas de importar y exportar, refactorización y autocompletado de código. También tienen como objetivo ayudar a los desarrolladores a crear aplicaciones web de forma rápida y sencilla. Está constituida por 4 módulos:

# *Fundamentación teórica de los procesos de desarrollo de software del módulo Configuración del Sistema de Gestión de Ensayos Clínicos*

---

**RichFaces VE:** editor visual que incluye soporte para las librerías de componentes JSF (Java Server Faces) incluyendo JBoss RichFaces. Brinda el apoyo para la edición visual de páginas HTML (Hypertext Markup Language), JSF, JSP (Java Server Pages) y Facelets.

- **Seam Tools:** incluye soporte para la vinculación de los componentes del framework Seam
- **Hibernate Tools:** soporta el mapeo de archivos, anotaciones y JPA (Java Persistence API) con ingeniería inversa, completamiento de código, asistentes de proyecto, refactorización, ejecución interactiva de HQL (Hibernate Query Language)/JPA-QL (JPA Query Language)/Criteria.
- **JBoss AS Tools:** contiene funciones para el despliegue eficiente de cualquier tipo de proyecto en el IDE.(13)

## **Visual Paradigm 6.4**

Es una herramienta de Ingeniería de Software Asistida por Computadora (CASE, por sus siglas en inglés) que utiliza UML para visualizar y diseñar elementos de software. Soporta el ciclo de vida completo del desarrollo de software: modelado de negocio, análisis y diseño orientados a objetos, implementación, pruebas y despliegue. El software de modelado UML ayuda a una rápida construcción de aplicaciones de calidad, mejores y a un menor coste, así como también permite dibujar todos los tipos de diagramas de clases y provee soporte para la generación de código. (14)

## **PostgreSQL 8.4**

Esta herramienta es un sistema de gestión de base de datos relacional orientada a objetos y libre, publicado bajo la licencia BSD (Berkeley Software Distribution). Es más completo que MySQL ya que permite métodos almacenados, restricciones de integridad y vistas. Puede ejecutarse sobre la mayoría de los sistemas operativos actuales. Incluye características orientadas a objetos, como puede ser la herencia, tipos de datos, funciones, restricciones, disparadores. A pesar de esto, PostgreSQL no es un sistema de gestión de bases de datos puramente orientado a objetos. (15)

## **1.8 Lenguajes de Programación**

### **Java**

Es un lenguaje de programación orientado a objetos. Fue desarrollado por la compañía Sun Microsystems, con la idea original de usarlo para la creación de páginas web. Permite el desarrollo de aplicaciones bajo el esquema de Cliente Servidor, como de aplicaciones distribuidas, lo que lo hace capaz

# *Fundamentación teórica de los procesos de desarrollo de software del módulo Configuración del Sistema de Gestión de Ensayos Clínicos*

---

de conectar dos o más computadoras u ordenadores, ejecutando tareas simultáneamente, y de esta forma logra distribuir el trabajo a realizar (16).

## **JavaScript**

Es un lenguaje de programación que se utiliza principalmente para crear páginas web dinámicas, los programas escritos con Java Script se pueden probar directamente en cualquier navegador sin necesidad de procesos intermedios. Este lenguaje es ligero e interpretado y orientado a objetos.(17)

### **1.9 Tecnologías utilizadas para desarrollo del sistema**

A continuación se sintetizarán las tecnologías necesarias que serán usadas para llevar a cabo la programación del módulo Configuración del sistema SIGEC 1.0, las cuales fueron definidas por el Departamento de Sistemas Especializados en Salud.

#### **1.9.1 Frameworks**

##### **Java Server Faces (JSF)**

Es un framework de desarrollo basado en el patrón MVC (Modelo-Vista-Controlador) para la creación de aplicaciones java de tipo web, que simplifica el desarrollo de interfaces de usuario en aplicaciones Java Enterprise Edition (Java EE, por sus siglas en inglés). Java Server Faces, es una tecnología potente, flexible, basada en componentes diseñados para simplificar el desarrollo Web en Java. También cuenta con excelentes herramientas y entornos de desarrollo de alta productividad. (18)

##### **Facelets**

Framework simplificado de presentación, es un lenguaje de declaración de páginas, poderoso pero ligero, que es usado para construir vistas de Java Server Faces usando plantillas de estilo de HTML (Hipertext Markup Language) y construyendo árboles de componentes. El uso de Facelets reduce el tiempo y esfuerzo que se necesita para el desarrollo y despliegue de aplicaciones. Así como también es un soporte para el Lenguaje de Expresiones (EL, por su siglas en inglés) y usa plantillas para componentes y páginas.

El mismo brinda soporte para reutilización de código por medio de plantillas y componentes compuestos, extensibilidad funcional de componentes y otros objetos del lado del servidor por configuración y tiempo de compilación rápido. (19)

# *Fundamentación teórica de los procesos de desarrollo de software del módulo Configuración del Sistema de Gestión de Ensayos Clínicos*

---

## **Java Persistence API**

Es un framework para el trabajo con bases de datos relacionales, compuesta por tres partes: el paquete compuesto por javax.persistence, el lenguaje de consultas de persistencia, los metadatos para los objetos y sus relaciones. El API Java Persistence fue desarrollado por el grupo de expertos de software de EJB 3.0 (Enterprise JavaBeans 3.0) como parte de JSR 220 (Java Specification Request), pero su uso no se limita a los componentes de software de EJB. También puede ser utilizado directamente por las aplicaciones web y clientes de la aplicación. (20)

## **JBoss Seam 2.1.1**

Es una potente plataforma de desarrollo de código abierto para crear aplicaciones de Internet en Java. Integra la capa de presentación JSF (Java Server Faces) con la capa de negocios y persistencia EJB (Enterprise JavaBeans). Con esta plataforma basta agregar anotaciones propias de éste a los objetos Entidad y Sesión de EJB, logrando con esto escribir menos código Java y XML (Extensible Markup Language). Otra característica importante es que permite hacer validaciones en los POJOs (Plain Object Java) como también manejar directamente la lógica de la aplicación y de negocios desde las sesiones de las controladoras. (21)

## **Hibernate 3.0**

Herramienta que facilita el mapeo de atributos entre una base de datos relacional tradicional y el modelo de objetos de una aplicación mediante archivos declarativos XML (Extensible Markup Language). Es una herramienta de Mapeo Objeto Relacional (ORM, por sus siglas en inglés) y un generador de sentencias SQL (Structured Query Language). Permite diseñar objetos persistentes que podrán incluir, relaciones, colecciones y un gran número de tipos de datos. Permite expresar consultas en una extensión de SQL (HQL), así como en SQL nativo o utilizando criterios orientado a objetos. Puede ser usado para desarrollar y distribuir aplicaciones de forma gratuita e implementa las especificaciones de EJB3 (Enterprise Java Beans). (22)

## **Enterprise Java Beans 3.0**

Es un componente de lado del servidor que encapsula la lógica del negocio de una aplicación. Es una plataforma para construir aplicaciones de negocios portables, reutilizables y escalables usando lenguaje de programación Java. La utilización de este componente aporta como beneficios: simplificar el desarrollo,

# *Fundamentación teórica de los procesos de desarrollo de software del módulo Configuración del Sistema de Gestión de Ensayos Clínicos*

---

el contenedor EJB (Enterprise Java Beans) es responsable de la administración de transacciones y autorizaciones de seguridad, puede residir en diferentes servidores y puede ser invocados por un cliente remoto. (23)

## **1.9.2 Librerías**

### **Ajax4JSF**

Es una librería de código abierto que se integra totalmente en la arquitectura de JSF (Java Server Faces) y extiende la funcionalidad de sus etiquetas dotándolas con tecnología Ajax de forma limpia y sin añadir código Java script. Mediante este framework se puede variar el ciclo de vida de una petición JSF, recargar determinados componentes de la página sin necesidad de recargarla por completo, realizar peticiones al servidor automáticas y controlar cualquier evento de usuario.(24)

### **RichFaces 3.3.1**

RichFaces es una librería de componentes visuales para JSF. Además, posee un framework avanzado para la integración de funcionalidades Ajax en estos componentes visuales, mediante el soporte de la librería Ajax4JSF. Se integra perfectamente en el ciclo de vida de JSF (Java Server Faces). Esta librería es un proyecto de código abierto. (25)

### **Seam UI**

Seam UI (Seam User Interface) es una serie de controles JSF altamente integrables con JBoss Seam. Adicionan varias mejoras a JSF, desde validación, expresiones extendidas e integración de la navegación en la interfaz de usuario.(26)

## **1.10 Tecnologías horizontales**

Las tecnologías horizontales son aquellas que se encuentran vinculadas a todas las capas que poseerá la arquitectura, a continuación se describen:

### **Java Enterprise Edition 5**

Esta plataforma se centra en hacer el desarrollo más fácil, sin embargo, conserva la riqueza de la plataforma J2EE 1.4. Ofrece características tales como la tecnología Java Server Faces APIs y servicios web. Java EE 5 hace que la codificación sea más simple y directa pero mantiene el poder que ha

# *Fundamentación teórica de los procesos de desarrollo de software del módulo Configuración del Sistema de Gestión de Ensayos Clínicos*

---

establecido Java EE como la primera plataforma para servicios web y desarrollo de aplicaciones empresariales. (27)

## **JBoss Application Server 4.2.2**

Es un servidor de aplicaciones J2EE de código abierto implementado en Java. Al estar basado en este lenguaje, JBoss puede ser utilizado en cualquier sistema operativo para el que esté disponible. Este servidor posee una serie de características entre las cuáles se encuentra: fue diseñado con una licencia de código abierto sin coste adicional, cumple los estándares, orientado a arquitectura de servicios y flexibilidad consistente. (28)

## **Java Virtual Machine**

La Máquina Virtual de Java (JVM, por sus siglas en inglés) es una computadora abstracta que ejecuta programas java compilados. La JVM es "virtual", ya que se lleva a cabo generalmente en el software en la parte superior de una verdadera plataforma de hardware y sistema operativo. Todos los programas de Java se compilan para la JVM. Por lo tanto, dicha máquina debe ser implementada en una plataforma en particular. (29)

Luego de la realización del presente capítulo se pudo comprobar que el HIS desarrollado por CESIM es actualmente el sistema del cual se podrán reutilizar funcionalidades de su módulo Configuración con el objetivo de disminuir el tiempo de desarrollo y garantizar una futura integración entre ambos sistemas. La utilización de la metodología RUP y el proceso de mejora CMMI, permitirá facilitar de acuerdo a las numerosas ventajas que presentan, el desarrollo de un producto con la calidad requerida. El estudio minucioso de las herramientas y tecnologías a utilizar, confirmó que mediante su utilización se obtendrán los beneficios necesarios para la implementación del módulo Configuración del sistema SIGEC 1.0.

# *Características del módulo Configuración del Sistema de Gestión de Ensayos Clínicos*

---

## **CAPÍTULO 2: CARACTERÍSTICAS DEL MÓDULO CONFIGURACIÓN DEL SISTEMA DE GESTIÓN DE ENSAYOS CLÍNICOS.**

Para llevar a cabo el desarrollo de cualquier sistema informático es necesario realizar un análisis de sus características para comprender los procesos existentes en el dominio del problema a resolver. En este capítulo se realiza una breve descripción de la propuesta de solución, así como una presentación visual de las clases conceptuales a través del modelo de dominio, además se especifican los requisitos funcionales y no funcionales del sistema y se definen los casos de uso para así conocer las funcionalidades con las que contará el módulo.

### **2.1 Propuesta de solución**

Luego de un profundo análisis de los sistemas existentes tanto a nivel nacional como internacional, que gestionan información sobre la configuración y administración, el sistema informático HIS es el único que cuenta con funcionalidades que se pueden integrar a la solución del problema planteado. De acuerdo a las disímiles potencialidades y a las características que presentan cada una de las aplicaciones restantes, se pudo comprobar que no son factibles para dar solución a la problemática identificada.

Por su parte el sistema Clínicas 1.0, que contiene la gran mayoría de las funcionalidades necesarias, para su construcción, no se utilizó un framework de desarrollo, por lo que el programador tenía que realizar las conexiones a la base de datos mediante códigos, y el servidor de aplicaciones Apache Tomcat 5.5 utilizado es obsoleto, comparado con la actual versión 7.0. Por otro lado, el soporte de este software de acuerdo a las características que posee hoy en día, fuera sumamente costoso y no sería factible incorporar las nuevas funcionalidades identificadas, pues daría como resultado un proceso más lento.

Es por ello que la creación del Sistema Informático para la Gestión de Ensayos Clínicos 1.0 (SIGEC), reutilizando funcionalidades del módulo Configuración del HIS, permitirá disminuir el tiempo de desarrollo, se ahorrará además una serie de pasos posibilitando el progreso de la aplicación y existirá la posibilidad de una futura integración con el sistema HIS.

SIGEC 1.0, además de brindar las funcionalidades presentes en Clínicas 1.0 permitirá: Habilitar y Deshabilitar centros, estudios y usuarios, así como Mostrar centros y estudios, Asignar rol a un usuario, Editar perfil.

# *Características del módulo Configuración del Sistema de Gestión de Ensayos Clínicos*

---

## **2.2 Modelo de Dominio**

Un modelo del dominio o modelo conceptual una representación visual de las clases conceptuales u objetos del mundo real en un dominio de interés. Se utiliza con frecuencia como fuente de inspiración para el diseño de los objetos software. El mismo muestra (a los modeladores) clases conceptuales significativas en un dominio de problema; es el artefacto más importante que se crea durante el análisis orientado a objetos. (30)

### **2.2.1 Conceptos fundamentales del dominio**

Para un mejor entendimiento del modelo del dominio, a continuación se realiza una breve descripción de algunos conceptos asociados al dominio del problema.

Estudio: es una investigación clínica realizada sobre pacientes, donde se recogen los datos generales del ensayo clínico.

Centro: institución donde se realizan los ensayos clínicos.

Rol: papel o cargo desempeñado por una determinada persona en la aplicación.

Usuario: persona que por medio de un ordenador puede acceder a los recursos y servicios que ofrece el sistema, de acuerdo con los privilegios, permisos y roles asignados.

Dirección IP (IP, es un acrónimo para Internet Protocol): es un número único e irrepetible con el cual se identifica una computadora conectada a una red.

Permiso: se refiere a la capacidad o el privilegio que tiene una persona al acceso a funcionalidades o a los distintos módulos del sistema.



# Características del módulo Configuración del Sistema de Gestión de Ensayos Clínicos

## 2.1.2 Diagrama del Modelo de Dominio

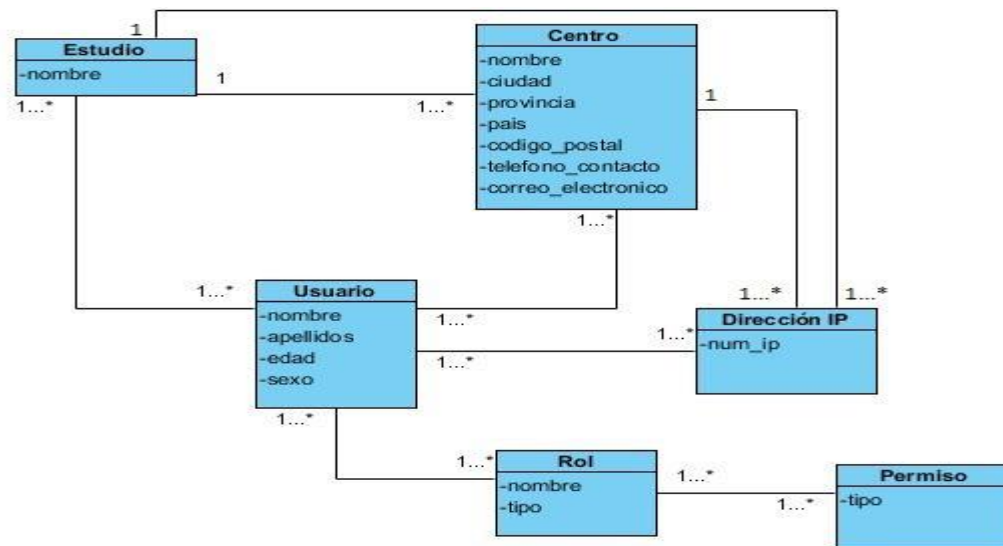


Fig. 1 Diagrama de Modelo de Dominio

Durante el proceso de configuración del sistema, un estudio tiene la posibilidad de estar asociado a varios centros y además tener asociado usuarios. Cada usuario puede estar inscrito en varios estudios y también tener asignado uno o disímiles roles a los que le son otorgados un conjunto de permisos en cada sección del sistema; así como las direcciones IP son asignadas a usuarios, centros y estudios.

## 2.3 Especificación de los requisitos del sistema (ERS)

Para el desarrollo de una aplicación informática que cumpla con las pautas definidas por el cliente final del software y se obtenga con la calidad requerida, se realiza un proceso de descripción de requisitos. La ERS, en esencia, es una descripción completa del comportamiento del sistema que se va a desarrollar, además está dirigida tanto al cliente como al equipo de desarrollo.

### 2.3.1 Requisitos funcionales del sistema

Característica requerida del sistema que expresa una capacidad de acción del mismo, una funcionalidad; generalmente expresada en una declaración en forma verbal (31). A continuación se exponen los siguientes requisitos funcionales identificados:

- RF1: Configurar funcionalidades.
- RF2: Adicionar usuario.
- RF3: Listar usuarios.

## *Características del módulo Configuración del Sistema de Gestión de Ensayos Clínicos*

---

- RF4: Buscar usuario.
- RF5: Ver detalles de usuario.
- RF6: Modificar datos de usuario.
- RF7: Eliminar usuario.
- RF8: Deshabilitar usuario.
- RF9: Habilitar usuario.
- RF10: Asignar rango o direcciones IP a usuario.
- RF11: Eliminar rango o direcciones IP a usuario.
- RF12: Asignar rango o direcciones IP a centro.
- RF13: Asignar rango o direcciones IP a estudio.
- RF14: Eliminar rango o direcciones IP a centro.
- RF15: Eliminar rango o direcciones IP a estudio.
- RF16: Adicionar estudio.
- RF17: Listar estudios.
- RF18: Buscar estudio.
- RF19: Ver detalles de estudio.
- RF20: Modificar datos de estudio.
- RF21: Eliminar estudio.
- RF22: Habilitar estudio.
- RF23: Deshabilitar estudio.
- RF24: Adicionar centro.
- RF25: Ver detalles de centro.
- RF26: Listar centros.
- RF27: Buscar centro.
- RF28: Eliminar centro.
- RF29: Habilitar centro.
- RF30: Deshabilitar centro.
- RF31: Modificar centro.
- RF32: Buscar rol.
- RF33: Asignar rol a usuario.

# *Características del módulo Configuración del Sistema de Gestión de Ensayos Clínicos*

---

RF34: Adicionar rol.

RF35: Listar roles.

RF36: Ver detalles de rol.

RF37: Modificar rol.

RF38: Eliminar rol.

RF39: Editar Perfil.

RF40: Eliminar rol asignado a usuario.

RF41: Modificar rol asignado a usuario.

## **2.3.2 Requisitos no funcionales del sistema**

Describen aspectos del sistema que son visibles por el usuario que no incluyen una relación directa con el comportamiento funcional del sistema, los mismos incluyen restricciones como el tiempo de respuesta (desempeño), la precisión, recursos consumidos y la seguridad. (32). Estos requisitos no funcionales se agrupan en las siguientes clasificaciones:

### ➤ **Requisitos de Usabilidad**

El sistema tiene una interfaz de fácil aprendizaje y ambiente sencillo porque sus funcionalidades básicas están desarrolladas para ser visibles y su acceso será mediante el clic, para que los usuarios inexpertos puedan familiarizarse rápidamente. Cuenta con las pautas de diseño definidas para aplicaciones web del CESIM, permitiendo al usuario obtener de manera uniforme y organizada la información del software.

### ➤ **Requisitos de Confiabilidad**

La aplicación tiene un sistema de trazas que registra en la base de datos las acciones realizadas por los usuarios. Se mantiene la seguridad y el control a nivel de usuario, garantizando el acceso de los mismos sólo a los niveles establecidos de acuerdo a la función que realizan. Las contraseñas pueden cambiarse sólo por el propio usuario o restablecerse por el administrador del sistema. Además ninguna información que se haya ingresado en la aplicación es eliminada físicamente de la BD, independientemente de que para el software no exista. El sistema informático permite la recuperación de la información de la base de datos a partir de los respaldos o salvadas realizadas.

# *Características del módulo Configuración del Sistema de Gestión de Ensayos Clínicos*

---

## ➤ **Requisitos de Eficiencia**

El servidor de aplicaciones debe soportar un aumento de usuarios concurrentes por minuto de 1 a 400, así como el sistema minimizará el volumen de datos en las peticiones y además optimizará el uso de recursos críticos como la memoria. Para ello se potenciará como regla guardar en la memoria caché datos y recursos de alta demanda.

## ➤ **Requisitos de Soporte**

Una vez terminada la aplicación se instalará en el CIM para realizar pruebas piloto del software y pruebas de despliegue. Se debe acceder al sistema desde cualquier plataforma, así como garantizar que el mismo sea compatible con el resto de los módulos que se realizan y además el sistema informático debe estar bien documentado de forma tal que el tiempo de mantenimiento sea mínimo en caso de necesitarse.

## ➤ **Requisitos de Restricciones de diseño**

El diseño de la aplicación será basado en la Metodología RUP con el uso del lenguaje de modelado UML, Visual Paradigm para el modelado de los artefactos que se generan, como lenguaje de programación Java y como Gestor de Base de Datos PostgreSQL.

## ➤ **Requisitos para la documentación de usuarios en línea y ayuda del sistema**

Se confeccionarán Manuales de Usuario para cada uno de los módulos que contiene el sistema como ayuda a los usuarios finales del mismo.

## **Requisitos de Interfaz**

Las páginas creadas no cargarán muchas imágenes y contarán con pocos colores y las principales tendrán información que servirá de guía al usuario. Se hace uso de simbología mediante íconos para indicar el estado de los elementos utilizados en el diseño. Además, los íconos permiten acceder a funcionalidades específicas.

## ➤ **Requisitos de Licencia**

Teniendo en cuenta que el producto Clínicas 1.0 es resultado de la modificación del software de código abierto OpenClinica el cual se encuentra publicado en Internet y presenta licencia LGPL (Licencia Pública General Reducida) y el producto SIGEC 1.0 es una segunda versión de Clínicas 1.0, el uso de esta

# *Características del módulo Configuración del Sistema de Gestión de Ensayos Clínicos*

---

licencia exige que todas las versiones mejoradas que se publiquen sean software libre, razón por la cual el producto SIGEC 1.0 será liberado bajo la licencia LGPL.

## ➤ **Requisitos de Software**

Para la instalación de la aplicación se debe disponer del sistema operativo Windows o GNU Linux, servidor de base de datos Postgres 8.4, navegador: Mozilla Firefox 3.0 o superior, servidor web: JBoss Application Server 4.2.2 y además debe estar instalada la máquina virtual de Java: java-6-openjdk para GNU Linux o la jdk-6u3 o superior para Windows.

## ➤ **Requisitos de Hardware**

Los ordenadores que son utilizados por los usuarios del sistema para acceder a la aplicación y operar la misma deben tener los siguientes requisitos de hardware: tipo de procesador: Intel Pentium IV o superior, velocidad del procesador: 3.0GHz o superior, memoria RAM: mínimo 1GB (recomendado) y el servidor de base de datos: debe poseer 1 a 3 TB de espacio disponible, pues el volumen de información es bastante grande y perdura en el tiempo hasta 15 años.

## ➤ **Requisitos de Estándares Aplicables**

Cumple con el modelo CMMI nivel 2 para la gestión de la calidad, gestión de configuración, gestión de requisitos; así como los lineamientos de calidad definidos por la universidad.

## ➤ **Requisitos de Seguridad**

El acceso al sistema tanto de entrada como de análisis de datos, debe estar sometido a un proceso de autenticación del usuario. Además las contraseñas deben tener más de 7 caracteres de longitud y tener un nivel medio de fortaleza y cada usuario tiene asignado uno o varios roles en el sistema. Cada rol definido tiene niveles de acceso a las distintas funcionalidades del sistema.

## **2.4 Modelo de casos de uso del sistema**

El modelo de casos de uso describe la funcionalidad propuesta del nuevo software, representa una unidad discreta de interacción entre un usuario (humano o máquina) y el sistema. Este modelo está conformado por actores, casos de uso y las relaciones que se establecen entre ellos.

# Características del módulo Configuración del Sistema de Gestión de Ensayos Clínicos

## 2.4.1 Definición de los actores del sistema

Actor se le denomina a toda entidad externa que demanda funcionalidad del sistema, es quien utiliza el sistema informático y envía o recibe de la aplicación mensajes o intercambia información con el software.(33)

Actor	Descripción
Administrador	Individuo que gestiona los datos de los usuarios en el sistema, los roles, direcciones IP y asigna permisos.
Usuario	Persona que gestiona en la aplicación todos los datos referentes a los estudios y centros.

Tabla 1. Definición de los actores del sistema

## 2.4.2 Diagrama de casos de uso del sistema

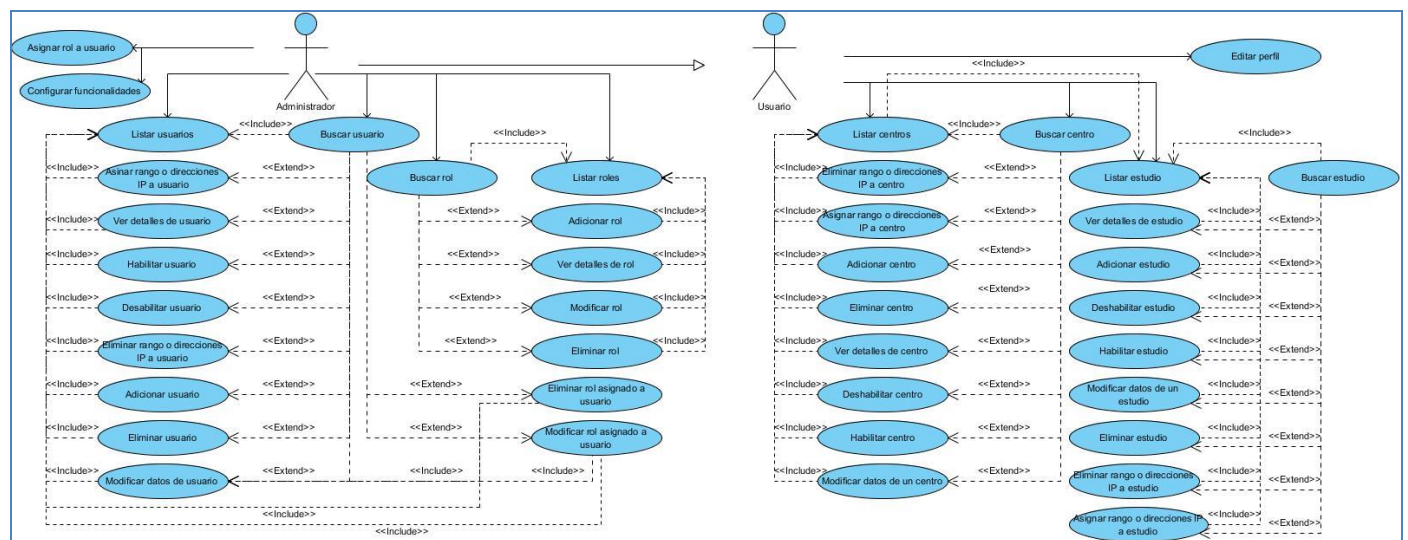


Fig. 2. Diagrama de casos de uso del sistema

## 2.4.3 Descripción de los casos de uso.

A continuación se describen algunos casos de uso que conforman el diagrama del subepígrafe anterior. Para más información referente a la descripción de los demás casos de uso, ver [Anexo1](#).

## *Características del módulo Configuración del Sistema de Gestión de Ensayos Clínicos*

---

<b>CU1: Listar centros</b>	
<b>Objetivo</b>	Mostrar al usuario el listado de los centros existentes en la aplicación.
<b>Actores</b>	Usuario
<b>Resumen</b>	El caso de uso inicia cuando el Usuario selecciona la opción Gestionar centros, el sistema muestra los centros existentes y las diferentes acciones asociadas a los mismos, termina el caso de uso.
<b>Referencias</b>	RF12: Asignar rango o direcciones IP a centro. RF14: Eliminar rango o direcciones IP a centro. RF17: Listar estudios. RF24: Adicionar centro. RF25: Ver detalles de centro. RF26: Listar centros. RF28: Eliminar centro. RF29: Habilitar centro. RF30: Deshabilitar centro.
<b>Flujo Normal de los eventos</b>	
<b>Actor</b>	<b>Sistema</b>

## *Características del módulo Configuración del Sistema de Gestión de Ensayos Clínicos*

---

1. Selecciona la opción “Gestionar estudios”	2. Muestra un listado de los estudios existentes en el sistema y las opciones: Ver, Modificar, Adicionar, Eliminar, Buscar, Habilitar, Deshabilitar, Gestionar direcciones IP estudios y Gestionar centros.
3. Selecciona la opción “Gestionar centros”	4. Muestra un listado de los centros existentes en el sistema y las opciones: Ver, Modificar, Adicionar, Eliminar, Buscar, Habilitar, Deshabilitar y Gestionar direcciones IP centros.
	5. El caso de uso termina.

Tabla 2.Descripción del caso de uso: *Listar centros*

<b>CU2: Asignar rango o direcciones IP a centro</b>	
<b>Objetivo</b>	Asigna rango o direcciones IP a un centro definido en la aplicación.
<b>Actores</b>	Administrador del sistema.



## *Características del módulo Configuración del Sistema de Gestión de Ensayos Clínicos*

---

<b>Resumen</b>	El caso de uso inicia cuando el Administrador selecciona la opción Gestionar direcciones IP centros, donde el sistema brinda la posibilidad de introducir y/o seleccionar los datos para asignar rango o direcciones IP al centro, el actor introduce los datos, el sistema asigna un rango o direcciones IP, el caso de uso termina.
<b>Referencias</b>	RF12: Asignar rango o direcciones IP a centro. RF26: Listar centros. RF27: Buscar centro
<b>Flujo Normal de los eventos</b>	
<b>Actor</b>	<b>Sistema</b>
1. Accede a la sección “Gestionar centros” de la opción “Gestionar estudios”	2. Muestra una interfaz con un listado de los centros existentes en el sistema y las opciones: Ver, Modificar, Adicionar, Eliminar, Buscar, Habilitar, Deshabilitar y Gestionar direcciones IP centros.
3. Selecciona la opción “Gestionar direcciones IP centros”	4. Muestra el campo Dirección IP con una acción para adiccionarla al centro, además de otro con el nombre de Rango de direcciones IP con su respectiva opción, así como un listado con los IP, rangos de IP con las acciones Eliminar IP, Eliminar rango de IP y el botón aceptar.

## *Características del módulo Configuración del Sistema de Gestión de Ensayos Clínicos*

---

5. Introduce la dirección IP o el rango de direcciones IP y presiona el botón Aceptar.	6. El caso de uso termina.
<b>Flujos alternos</b>	
Alternativa 1. "Datos incorrectos o incompletos"	
<b>Actor</b>	<b>Sistema</b>
	<ol style="list-style-type: none"> <li>1. Muestra una ventana con un mensaje de Error.</li> <li>2. Regresa al paso 4 del <b>Flujo Normal de los Eventos</b>.</li> </ol>
<b>Puntos de extensión</b>	CU. Buscar centro

Tabla 3.Descripción del caso de uso: *Asignar rango o direcciones IP a centro*

<b>CU3: Adicionar centro</b>	
<b>Objetivo</b>	Crear un nuevo centro en el sistema.
<b>Actores</b>	Usuario
<b>Resumen</b>	El caso de uso inicia cuando el Usuario accede a la acción Adicionar centro de la sección "Gestionar centros" donde el sistema brinda la posibilidad de introducir y/o seleccionar los datos para adicionar el nuevo centro, el actor introduce los datos, el sistema adiciona el centro, el caso de uso termina.

## *Características del módulo Configuración del Sistema de Gestión de Ensayos Clínicos*

---

<b>Referencias</b>	RF24: Adicionar centro. RF26: Listar centros. RF27: Buscar centro.
<b>Flujo Normal de los eventos</b>	
<b>Actor</b>	<b>Sistema</b>
1. Accede a la sección “Gestionar centros” de la opción “Gestionar estudios” del módulo configuración.	2. Muestra una interfaz con un listado de los centros existentes en el sistema y las opciones: Ver, Modificar, Adicionar, Eliminar, Buscar, Habilitar, Deshabilitar y Gestionar direcciones IP de los centros.
3. Selecciona la funcionalidad Adicionar centro.	4. Muestra los siguientes campos: Nombre del centro para crear el centro, ID único del protocolo, investigador principal, sumario breve, fecha de inicio, fecha estimada de finalización, número esperado de inclusiones, nombre de la instalación, ciudad/municipio del centro, estado/provincia de la instalación, país del centro, nombre de contacto en el centro, titulación del contacto en el centro, teléfono de contacto en el centro y correo electrónico de contacto en el centro, además el botón Aceptar y Cancelar.

## *Características del módulo Configuración del Sistema de Gestión de Ensayos Clínicos*

---

5. Introduce los datos especificados y presiona el botón Aceptar.	6. Adiciona el centro. El caso de uso termina.
<b>Flujos alternos</b>	
Alternativa 1. "Cancelar"	
<b>Actor</b>	<b>Sistema</b>
1. Selecciona la opción "Cancelar".	2. Regresa a la vista anterior. 3. El caso de uso termina.
Alternativa 2. "Datos incorrectos o incompletos"	
<b>Actor</b>	<b>Sistema</b>
	1. Muestra una ventana con un mensaje de Error. 2. Regresa al paso 4 del <b>Flujo Normal de los Eventos</b> .
<b>Puntos de extensión</b>	Buscar centro

Tabla 4. Descripción del caso de uso: *Adicionar centro*

Con el desarrollo de este capítulo se pudo alcanzar un mejor entendimiento sobre el alcance del software y las restricciones que deben existir para satisfacer las necesidades del cliente final. La confección de varios artefactos permitió enmarcar los conceptos asociados al dominio del problema y dar inicio al diseño del sistema propuesto.

# *Diseño del módulo Configuración del Sistema de Gestión de Ensayos Clínicos*

---

## **CAPÍTULO 3: DISEÑO DEL MÓDULO CONFIGURACIÓN DEL SISTEMA DE GESTIÓN DE ENSAYOS CLÍNICOS**

Para el modelado del módulo Configuración del sistema SIGEC 1.0, se tienen en cuenta varios artefactos importantes, como el diagrama de paquetes y el diagrama de clases del diseño. Se fundamenta la arquitectura empleada, se brinda además una descripción de las principales clases del diseño empleadas en el desarrollo de este subsistema.

### **3.1 Descripción de la arquitectura**

Una arquitectura de software es parte del diseño del sistema, donde se definen la estructura y las propiedades globales de la aplicación, incluye sus componentes, las propiedades observables de dichos componentes y las relaciones que se establecen entre ellos. Tratando además, aspectos del diseño y desarrollo que no pueden tratarse adecuadamente dentro de los módulos que forman el sistema informático (34). El módulo propuesto presenta la siguiente arquitectura.

#### **3.1.1 Arquitectura cliente-servidor**

Esta arquitectura consiste básicamente en que un programa cliente realiza peticiones a otro programa (el servidor) que le da respuesta. La interacción cliente-servidor es el soporte de la mayor parte de la comunicación por redes. Ayuda a comprender las bases sobre las que están contruidos los algoritmos distribuidos. Así como también la relación establecida puede ser de muchos a uno, en la que un servidor puede dar servicio a muchos clientes, regulando su acceso a los recursos compartidos. Unas de las ventajas que brinda dicha arquitectura es que facilita la integración entre sistemas diferentes y comparte información, permitiendo por ejemplo que las máquinas ya existentes puedan ser utilizadas mediante interfaces más amigables para el usuario. De esta manera, se puede integrar a computadoras personales con sistemas medianos y grandes, sin necesidad de que todos tengan que utilizar el mismo sistema operativo. (12)

#### **3.1.2 Patrón arquitectónico: Modelo Vista Controlador (MVC)**

El patrón de arquitectura MVC (Modelo Vista Controlador) es un patrón de arquitectura de las aplicaciones software que separa la lógica de negocio de la interfaz de usuario, así como también facilita la evolución por separado de ambos aspectos e incrementa reutilización y flexibilidad. Define la organización

# *Diseño del módulo Configuración del Sistema de Gestión de Ensayos Clínicos*

---

independiente del Modelo (Objetos de Negocio), la Vista (interfaz con el usuario u otro sistema) y el Controlador (controlador del flujo de datos de la aplicación). (35)

Este patrón tiene entre sus principales ventajas proporcionar un mecanismo de configuración para componentes complejos mucho más tratable que el puramente basado en eventos (el modelo puede verse como una representación estructurada del estado de la interacción), además posibilita obtener diferentes vistas para un mismo modelo y es posible construir nuevas vistas sin necesidad de modificar el modelo subyacente.

**Modelo:** representa las estructuras de datos. Típicamente el modelo de clases contendrá funciones para consultar, insertar y actualizar información de la base de datos.

**Vista:** es la información presentada al usuario. Una vista puede ser una página web o una parte de ella.

**Controlador:** actúa como intermediario entre el modelo, la vista y cualquier otro recurso necesario para generar una página web. (36)

Esta arquitectura descrita anteriormente será la utilizada debido a las numerosas ventajas que presenta para la implementación del módulo Configuración, y además es la definida para el desarrollo del sistema.

## **3.2 Modelo de diseño**

Es un modelo de objetos que describe la realización de los casos de uso en el dominio de la solución, está compuesto por artefactos que engloban todas las clases del diseño, subsistemas, paquetes, colaboraciones, y las relaciones entre ellos.

Mediante el modelo de diseño se hace un refinamiento del proceso de análisis anteriormente realizado. Se centra principalmente en cómo los requisitos funcionales y no funcionales, junto a otras restricciones relacionadas con el entorno de implementación, tienen impacto en el sistema, por lo que constituye un plano del modelo de implementación. Es usado como una entrada inicial en las actividades de implementación y prueba.

En esta fase se elaboran una serie de artefactos entre ellos el diagrama de clases del diseño, donde se exponen un conjunto de interfaces, controladoras y las relaciones entre las mismas, así como el diagrama de paquetes, el cual define una estructura de paquetes que permiten dividir el sistema en fragmentos manejables para su implementación.

# *Diseño del módulo Configuración del Sistema de Gestión de Ensayos Clínicos*

---

## **3.2.1 Patrones de diseño**

Los patrones de diseño brindan una solución ya probada y documentada a problemas comunes en el desarrollo de software, que están sujetos a contextos similares. Entre los patrones de diseño que se utilizan se encuentran los patrones GRASP (General Responsibility Assignment Software Patterns, por sus siglas en inglés), los cuales describen los principios fundamentales de la asignación de responsabilidades a objetos, expresados en formas de patrones (38). Los patrones Experto, Creador, Controlador, Alta Cohesión y Bajo Acoplamiento son los utilizados en el presente trabajo de diploma.

**Patrón Experto:** es el principio básico de la asignación de responsabilidades, el mismo asigna una responsabilidad a la clase que tiene la información necesaria para cumplirla. Este se evidencia en la clase controladora `CrearEstudio_disenno`, donde se crea un nuevo estudio, utilizando la información proporcionada por la entidad estudio.

**Patrón Creador:** permite creación de nuevas instancias por las clases que tienen la información necesaria para realizar la confección del objeto, usa directamente las instancias creadas del objeto, almacena o maneja varias instancias de la clase y contiene o agrega la clase. Este patrón se refleja en la controladora `CrearEstudio_disenno` la cual crea una instancia u objeto de la clase `Study`.

**Patrón Controlador:** se aplica para asignar la responsabilidad de gestionar un mensaje de un evento del sistema a una clase que representa el sistema global, dispositivo o subsistema (controlador de fachada) o representa un escenario de caso de uso en el que tiene lugar el evento del software (controlador de caso de uso o de sesión). El mismo se pone de manifiesto en la creación de una clase controladora por cada caso de uso de la aplicación.

**Patrón Alta Cohesión:** se utiliza para asignar responsabilidades de manera que la información que almacena una clase sea coherente y esté relacionada con la clase. Este se evidencia en la clase controladora `CrearEstudio_disenno`.

**Patrón Bajo Acoplamiento:** es la idea de tener las clases lo menos ligadas entre sí, de tal forma que en caso de producirse una modificación en alguna de ellas, se tenga la mínima repercusión posible en el resto de las clases, potenciando la reutilización y disminuyendo la dependencia entre ellas. Dicho patrón se pone de manifiesto en la clase `CrearCentro_diseño`.

# *Diseño del módulo Configuración del Sistema de Gestión de Ensayos Clínicos*

---

Los patrones de diseño **GoF** (por sus siglas en inglés, Gang of Four) permite reducir el acoplamiento entre los objetos y se clasifican en tres grandes categorías basadas en su propósito: creacionales, estructurales y de comportamiento. En la presente investigación se utiliza el patrón de comportamiento, el cual ayuda a definir la comunicación e iteración entre los objetos de un software. Dentro de este patrón se utiliza el Iterador, que permite realizar recorridos sobre objetos compuestos independientemente de la implementación de estos. Se evidencia cuando se cargan las listas en la aplicación y se muestran los datos de las mismas.

### **3.2.2 Diagrama de paquetes**

Los paquetes constituyen un mecanismo utilizado para agrupar elementos de UML. Por su parte el diagrama de paquetes muestra como un sistema está dividido en agrupaciones lógicas mostrando las dependencias entre esas agrupaciones. Dado que normalmente un paquete está pensado como un directorio, los diagramas de paquetes suministran una descomposición de la jerarquía lógica de un sistema. Permiten dividir un modelo para agrupar y encapsular sus elementos en unidades lógicas individuales.

El paquete que engloba los 3 subpaquetes llamado Repositorio de clases está compuesto por:

**Entidades:** está formado por las clases autogeneradas especificadas en el diseño, en dependencia de las tecnologías usadas en la implementación y las personalizadas. Estas clases se autogeneran desde la base de datos utilizando el ORM Hibernate (generación de objetos java desde la base de datos) y personalizadas son aquellas que se modifican para una mejor gestión de la información, por lo que pueden heredar de las autogeneradas.

**Sesiones:** está conformado por las clases controladoras autogeneradas por el entorno de desarrollo, por las clases controladoras personalizadas y por las controladoras del proceso.

**Vistas:** contiene los contenidos web referentes a las páginas clientes y los formularios que las componen, además de las vistas que interactúan con el usuario.



# Diseño del módulo Configuración del Sistema de Gestión de Ensayos Clínicos

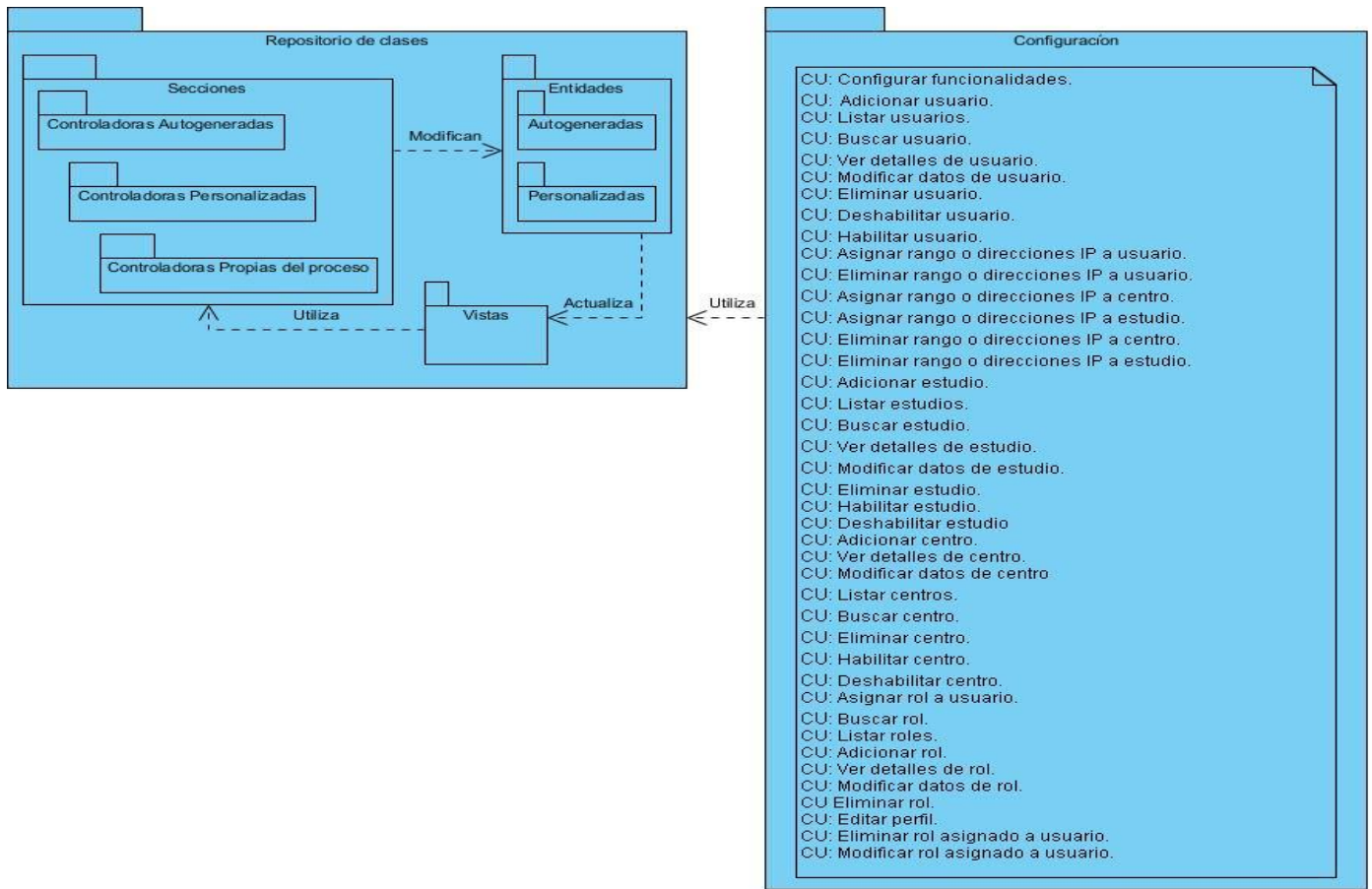


Fig. 3. Diagrama de paquetes

### 3.2.3 Diagrama de clases del diseño

El Diagrama de clases de diseño (DCD) describe gráficamente las especificaciones de las clases del software y las interfaces en una aplicación. Normalmente contiene la siguiente información: clases, asociaciones y atributos, interfaces con sus operaciones y constantes, así como información sobre los tipos de atributos, además de navegabilidad y dependencias (39). Cada uno de ellos tiene asociado un diagrama de secuencia, los cuales muestran la forma en que un grupo de objetos se comunican (interactúan) entre sí a lo largo del tiempo. Estos constan de objetos, mensajes entre estos objetos y una línea de vida del objeto representada por una línea vertical (40). A continuación se reflejará una breve descripción de los componentes que conforman dicho DCD.

# Diseño del módulo Configuración del Sistema de Gestión de Ensayos Clínicos

---

## Descripción de clases del diseño

**Página cliente:** representan las interfaces del sistema. Utilizan un conjunto de librerías y pueden contener código Java Script para la validación de los datos.

**Página servidora:** encargadas de construir las interfaces (páginas clientes) del sistema.

**Formulario:** Un formulario HTML es una sección de un documento enmarcado entre etiquetas <form>, que contiene elementos llamados controles como casillas de verificación (checkboxes), radiobotones (radiobuttons), menús, y rótulos (labels). Estos formularios son completados por los usuarios modificando sus controles (introduciendo texto, seleccionando objetos de un menú), los cuales son enviados al servidor para su procesamiento.

**Controladoras:** se crea una clase controladora por cada caso de uso, encargada de la ejecución de cada una de las funcionalidades relacionadas con el módulo. Contiene los métodos que implementan los requisitos funcionales. Invoca a la clase SEAM (Servlet) propia del framework SEAM.

Nombre: CrearCentro_disenno	
Tipo de clase: Controladora	
Atributo	Tipo
Centro	Study
estudioPadre	Study
studyId	long
parentStudyId	BigDecimal
uniqueIdentifier	String
Name	String
Summary	String
datePlannedStart	Date
datePlannedEnd	Date
dateCreated	Date
statusId	BigDecimal
principalInvestigator	String

## *Diseño del módulo Configuración del Sistema de Gestión de Ensayos Clínicos*

---

facilityName	String
facilityCity	String
facilityState	String
facilityZip	String
facilityCountry	String
facilityContactName	String
facilityContactDegree	String
facilityContactPhone	String
facilityContactEmail	String
expectedTotalEnrollment	BigDecimal
<b>Para cada responsabilidad:</b>	
<b>Nombre</b>	List<Study>ListadodeCentros()
<b>Descripción</b>	Muestra un listado de los centros existentes en la aplicación.

## *Diseño del módulo Configuración del Sistema de Gestión de Ensayos Clínicos*

---

<b>Nombre</b>	String ObtenerOcid()
<b>Descripción</b>	Devuelve el (ocid) que se le va a agregar al centro, una vez creado.
<b>Nombre</b>	crearCentro()
<b>Descripción</b>	Crea un nuevo centro en el sistema.

Tabla 5.Descripción de la clase controladora: *CrearCentro\_disenno*

<b>Nombre: CrearDireccion_ip_administracion</b>	
<b>Tipo de clase:</b> Controladora	
Atributo	Tipo
studyIp	Study
Ip	String
iniciorangoA	String
finalrangoA	String
<b>Para cada responsabilidad:</b>	
<b>Nombre</b>	String[] MostrarListaIP(String ip1)
<b>Descripción</b>	En la base de datos los ip aparecen en un solo campo unidos y esta funcionalidad permite devolver una lista con los ip separados.
<b>Nombre</b>	String[] MostrarListaRangoIP(String rangoip)
<b>Descripción</b>	En la base de datos los rangos aparecen en un solo campo unidos y esta funcionalidad posibilita devolver una lista con los rangos separados.
<b>Nombre</b>	void AgregarIPListaIP()

## *Diseño del módulo Configuración del Sistema de Gestión de Ensayos Clínicos*

---

<b>Descripción</b>	Adiciona un ip a la lista de ip y a la base de datos del centro o del estudio.
<b>Nombre</b>	boolean CompararIpMenor(String direccion1[], String direccion2[])
<b>Descripción</b>	Compara dos direcciones ip y devuelve la menor.
<b>Nombre</b>	int CompararIpMenorParaRango(String direccion1[], String direccion2[])
<b>Descripción</b>	Compara dos direcciones ip y devuelve la menor o igual.
<b>Nombre</b>	List<String> OrdenarListaMostrarRangos()
<b>Descripción</b>	Ordena la lista de rangos ip y la modifica en la base de datos.
<b>Nombre</b>	List<String> OrdenarListaMostrar()
<b>Descripción</b>	Ordena la lista de ip y la modifica en la base de datos.
<b>Nombre</b>	void AgregarRangoListaRango()
<b>Descripción</b>	Adiciona un rango ip a la lista de ip y a la base de datos.
<b>Nombre</b>	void EliminarIP(String ipEliminar)
<b>Descripción</b>	Elimina un ip de la lista de ip y de la base de datos.
<b>Nombre</b>	void EliminarRangoIP(String rangoIpEliminar)
<b>Descripción</b>	Elimina un rango ip de la base de datos.

Tabla 6. Descripción de la clase controladora: *CrearDireccion\_ip\_administracion*

Nombre: Study	
Tipo de clase: Entidad	
Atributo	Tipo
studyId	Long
parentStudyId	BigDecimal

## *Diseño del módulo Configuración del Sistema de Gestión de Ensayos Clínicos*

---

uniqueIdentifier	String
secondaryIdentifier	String
Name	String
Summary	String
datePlannedStart	Date
datePlannedEnd	Date
dateCreated	Date
dateUpdated	Date
ownerId	Bigdecimal
updateId	Bigdecimal
Attributeattribute	Bigdecimal
statusId	Bigdecimal
principalInvestigator	String
facilityName	String
facilityCity	String
facilityState	String
facilityZip	String
facilityCountry	String
facilityRecruitmentStatus	String
facilityContactName	String
facilityContactDegree	String
facilityContactPhone	String
facilityContactEmail	String
protocolType	String
protocolDescription	String
protocolDateVerification	Date
Phase	String

## *Diseño del módulo Configuración del Sistema de Gestión de Ensayos Clínicos*

---

expectedTotalEnrollment	BigDecimal
Sponsor	String
Collaborators	String
medlineIdentifier	String
url	String
urlDescription	String
Conditions	String
Keywords	String
Eligibility	String
Gender	String
ageMax	String
ageMin	String
healthyVolunteerAccepted	Boolean
Purpos	String
Allocation	String
Masking	String
Control	String
Assignment	String
Endpoint	String
Nterventions	String
Duration	String
Selection	String
Timing	String
officialTitle	String
resultsReference	String
ocOid	String
lp	String

## *Diseño del módulo Configuración del Sistema de Gestión de Ensayos Clínicos*

---

ipRango	String
secondaryStatus	BigDecimal
Cronogramas	Set<Cronograma>
eventDefinitionCrfs	Set<EventDefinitionCrf>
studyParameterValues	Set<StudyParameterValue>
<b>Para cada responsabilidad:</b>	
<b>Descripción</b>	En esta entidad se insertan los datos de centros y estudios después de creados. También para mostrar el listado de estudios y centros, se cargan los datos a ilustrar de dicha tabla, así como otras funcionalidades con respecto a centros y estudios.

Tabla 7. Descripción de la clase entidad: *Study*

A continuación se exponen algunos diagramas de clases del diseño, pertenecientes a los casos de uso: Listar centros, Asignar rango o direcciones IP a centro, Adicionar centro. Para más información referente a los diagramas de los demás casos de uso, ver [Anexo 2](#).



# Diseño del módulo Configuración del Sistema de Gestión de Ensayos Clínicos

## DCD\_Listar centros

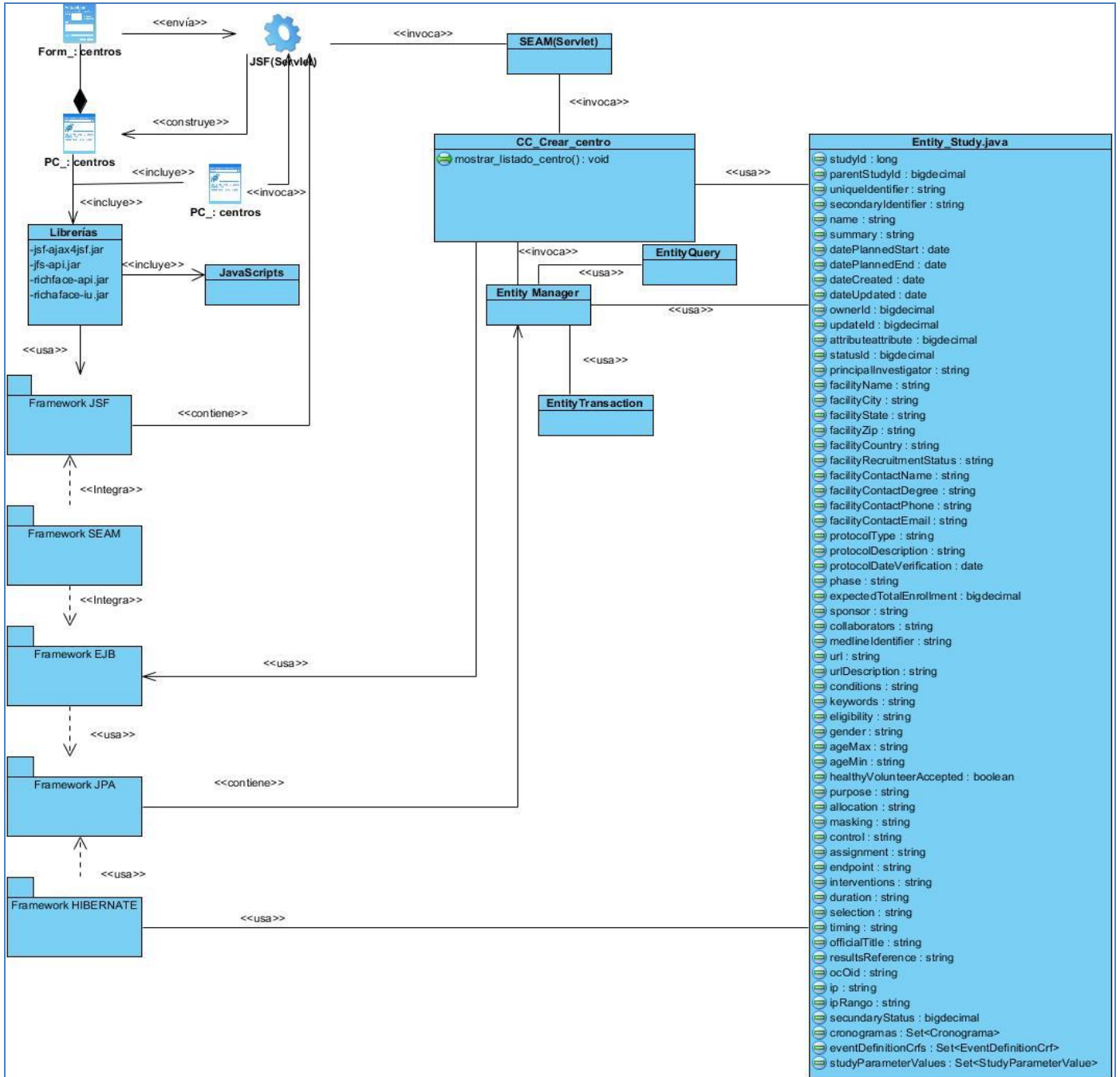


Fig. 4. Diagrama de clases del diseño: Listar centros

# Diseño del módulo Configuración del Sistema de Gestión de Ensayos Clínicos

## DCD\_ Asignar rango o direcciones IP a centro

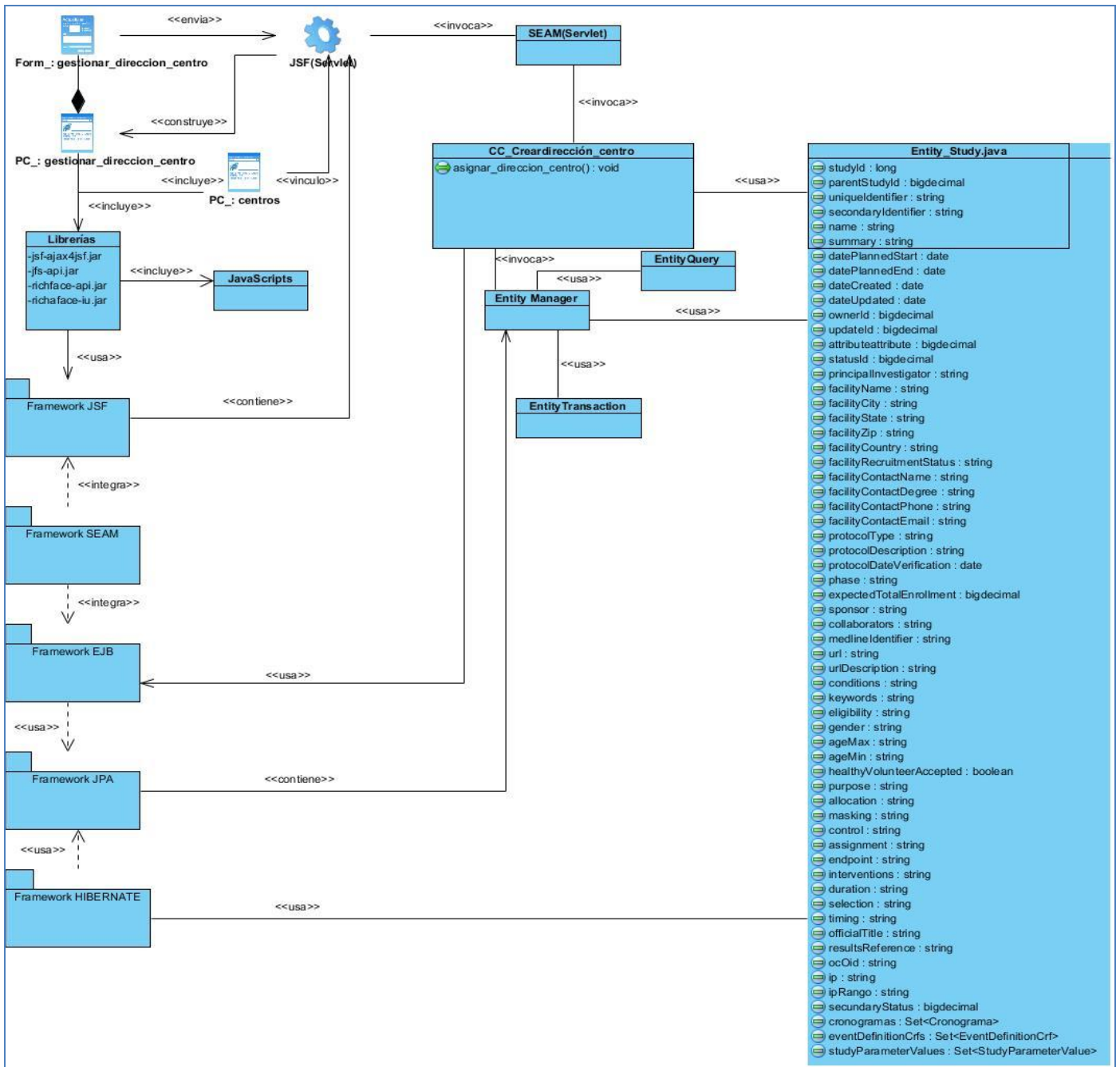


Fig. 5. Diagrama de clases del diseño: *Asignar rango o direcciones IP a centro*

# Diseño del módulo Configuración del Sistema de Gestión de Ensayos Clínicos

## DCD\_ Adicionar centro

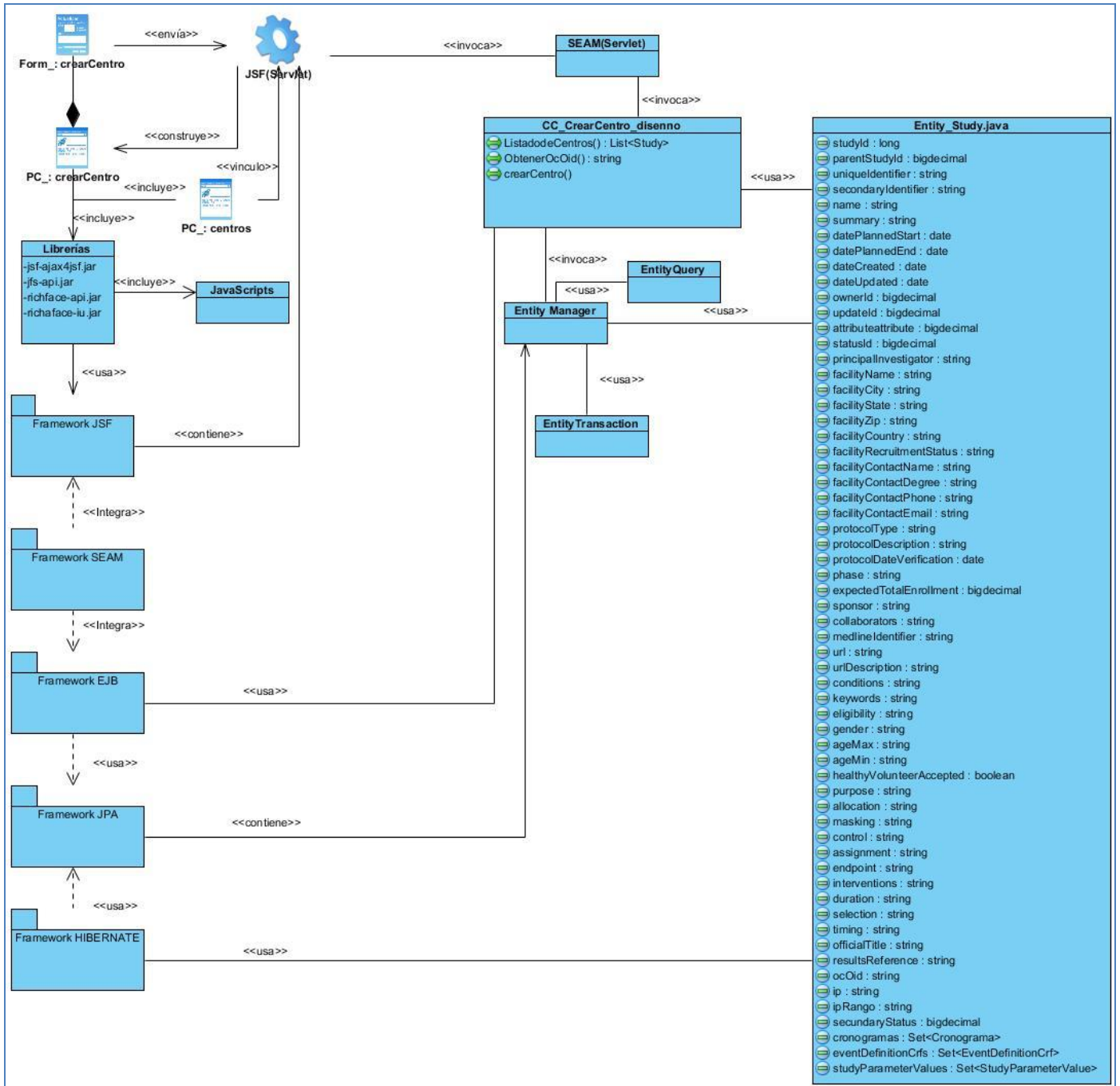


Fig. 6. Diagrama de clases del diseño: *Adicionar centro*

## *Diseño del módulo Configuración del Sistema de Gestión de Ensayos Clínicos*

---

Luego de la realización de este capítulo se pudo obtener el conocimiento necesario para llevar a cabo una implementación organizada a través de la utilización del patrón Modelo Vista Controlador. Se obtuvo además una mejor visión de cómo el sistema está organizado y dividido en agrupaciones lógicas visualizando e identificando las clases fundamentales que deben ser definidas para que el software funcione satisfactoriamente.

# *Implementación del módulo Configuración del Sistema de Gestión de Ensayos Clínicos*

---

## **CAPÍTULO 4: IMPLEMENTACIÓN DEL MÓDULO CONFIGURACIÓN DEL SISTEMA DE GESTIÓN DE ENSAYOS CLÍNICOS**

En el presente capítulo se muestra la implementación del sistema propuesto a raíz de los resultados obtenidos en el diseño. Se elabora el modelo de datos y se describen las tablas que lo integran, además se obtienen los diagramas de componentes y despliegue, así como las estrategias de codificación, estándares y el estilo de código a utilizar.

### **4.1 Modelo de Datos**

Un modelo de datos es la combinación de una colección de estructuras de datos, operadores o reglas de inferencia y de reglas de integridad, las cuales definen un conjunto de estados consistentes. El cual puede ser usado como una herramienta para especificar los tipos de datos y la organización de los mismos. Además para la manipulación de consultas y datos es el elemento clave en el diseño de la arquitectura de un manejador de base de datos.

Representa la combinación de tres componentes: una colección de estructuras de datos (los bloques constructores de cualquier base de datos que conforman el modelo), un conjunto de operadores o reglas de inferencia, para consultar o derivar datos de cualquier parte de estas estructuras en cualquier combinación deseada y una colección de reglas generales de integridad, las cuales explícita o implícitamente definen un conjunto de estados consistentes, estas reglas algunas veces son expresadas como reglas de insertar-actualizar-borrar. (41)

A continuación se muestra el modelo de datos del módulo Configuración, siendo sus elementos esenciales las entidades, los atributos y las relaciones entre las entidades.

# Implementación del módulo Configuración del Sistema de Gestión de Ensayos Clínicos

## 4.1.1 Diagrama de Modelo de Datos

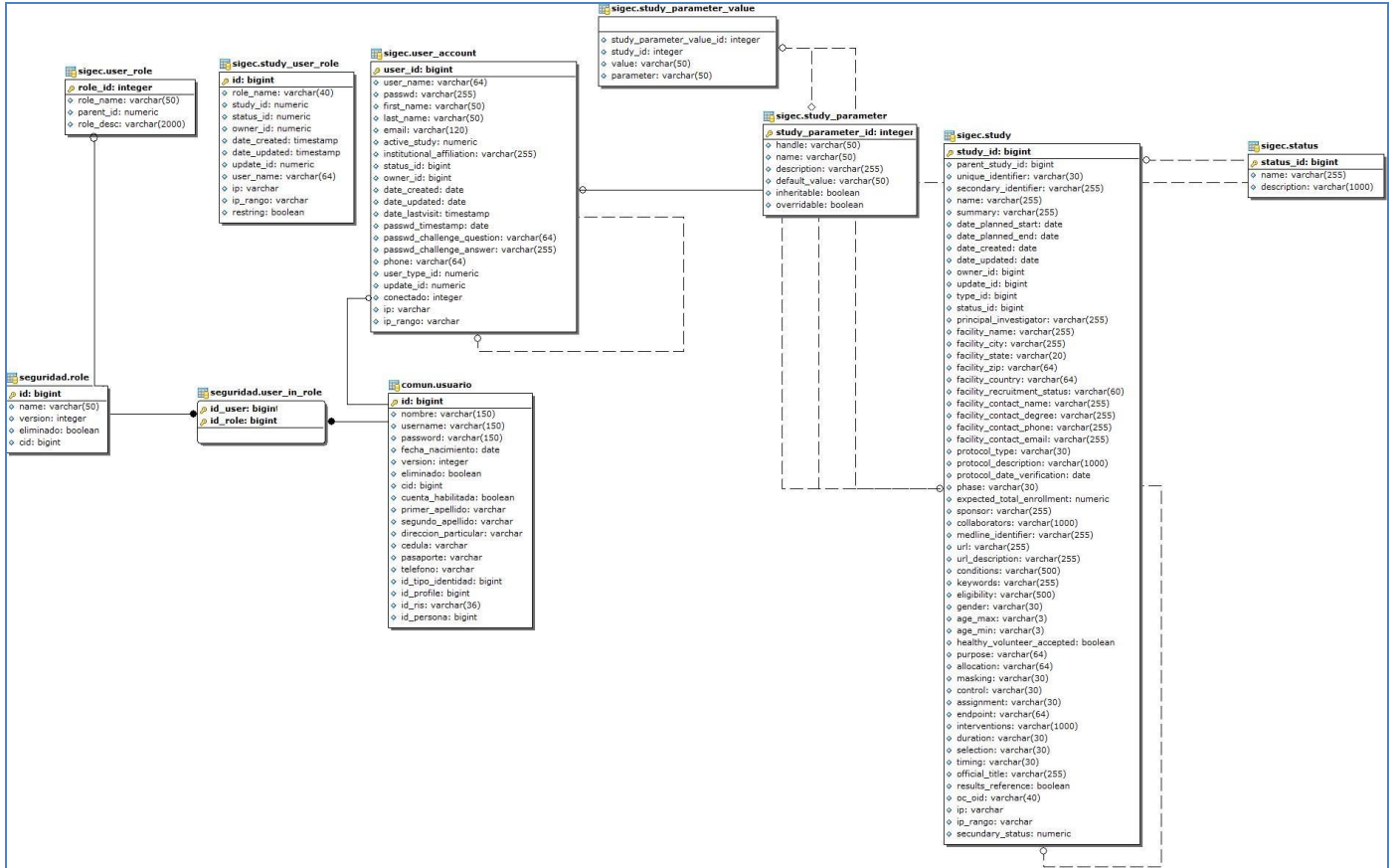


Fig. 7. Modelo de Datos

## 4.1.2 Descripción de las tablas de la base de datos

Nombre: study		
<b>Descripción</b>	En esta entidad se insertan los datos de centros y estudios luego de creados, también para mostrar el listado de estudios y centros se cargan los datos a ilustrar de dicha tabla, así como otras funcionalidades con respecto a centros y estudios.	
<b>Atributo</b>	<b>Tipo</b>	<b>Descripción</b>



## *Implementación del módulo Configuración del Sistema de Gestión de Ensayos Clínicos*

---

study_id	Bigint	Representa el identificador único del estudio.
parent_study_id	Bigint	Indica si el estudio tiene un padre (si este campo tiene valor es un centro).
unique_identifier	varchar	Representa el Id único del protocolo del estudio.
Name	varchar	Representa el título abreviado del estudio.
Summary	varchar	Representa el resumen con respecto al estudio.
date_planned_start	date	Fecha de comienzo del estudio.
date_planned_end	date	Fecha de finalización del estudio.
date_created	date	Día de la creación del estudio.
date_updated	date	Día de la actualización del estudio.
update_id	bigint	Representa la entidad que realizó la actualización del estudio.
status_id	bigint	Estado en el que se encuentra el estudio.
principal_investigator	varchar	Nombre del investigador principal que realiza el estudio.
facility_name	varchar	Nombre de la instalación donde se realiza el estudio.
facility_city	varchar	Nombre de la ciudad / municipio del centro donde se realiza el estudio.
facility_state	varchar	Nombre del estado / provincia de la instalación donde se realiza el estudio.
facility_zip	varchar	Código postal del centro donde se realiza el estudio.
facility_country	varchar	País del centro donde se realiza el estudio.
facility_contact_name	varchar	Nombre del contacto en el centro donde se realiza el estudio.
facility_contact_degree	varchar	Titulación del contacto en el centro donde se realiza el estudio.
facility_contact_phone	varchar	Teléfono del contacto en el centro donde se realiza el estudio.
facility_contact_email	varchar	Correo electrónico del contacto en el centro donde se realiza el estudio.

## *Implementación del módulo Configuración del Sistema de Gestión de Ensayos Clínicos*

---

protocol_type	varchar	Representa el tipo de protocolo: observacional o intervencionista.
protocol_description	varchar	Descripción detallada del protocolo seleccionado.
Phase	varchar	Representa la fase en la que se encuentra el estudio.
expected_total_enrollment	varchar	Representa el número esperado de inclusiones del estudio.
Sponsor	varchar	Nombre del patrocinador del estudio.
Collaborators	varchar	Nombre de los colaboradores del estudio.
medline_identifier	varchar	Identificador de MEDLINE.
url	varchar	URL (Uniform Resource Locator, por sus siglas en inglés) de referencia del estudio.
url_description	varchar	Descripción de la URL del estudio.
Conditions	varchar	Patología del estudio.
Keywords	varchar	Palabras claves (separadas por coma) del estudio.
Eligibility	varchar	Criterios de elegibilidad del estudio.
Gender	varchar	Sexo del sujeto para la realización del estudio.
age_max	varchar	Edad máxima de los sujetos del estudio.
age_min	varchar	Edad mínima de los sujetos del estudio.
ealthy_volunteer_accepted	boolean	Representa el valor (si o no) existen voluntarios sanos para el estudio.
Purpose	varchar	Representa si es un tratamiento, prevención diagnóstico o educacional/asesoramiento.
Allocation	varchar	Representa si es un estudio clínico con asignación aleatoria o con asignación no aleatoria.
Masking	varchar	Representa el tipo de enmascaramiento: abierto, ciego simple o doble ciego.
Control	varchar	Contiene el tipo de control: placebo, activo, sin controles, histórico o comparación de datos.
Assignment	varchar	Representa el tipo de asignación: grupo único, paralelo, cruzado o factorial.



## *Implementación del módulo Configuración del Sistema de Gestión de Ensayos Clínicos*

---

Endpoint	varchar	Punto final: seguridad, eficacia, seguridad/eficacia, bio-equivalencia, bio-disponibilidad, farmacocinética, farmacodinámica, farmacocinética / farmacodinámica.
Interventions	varchar	Contiene el nombre del propósito (Ej. medicamento dipirona).
Duration	varchar	Período de duración del estudio.
Selection	varchar	Este campo representa el tipo de selección: muestra por conveniencia, población definida, muestra aleatoria, control de caso.
Timing	varchar	Ajuste temporal del estudio.
official_title	varchar	Título oficial del estudio.
results_reference	boolean	Referencias de los resultados del estudio realizado.
oc_oid	varchar	Indica el identificador único del campo unique_identifier.
Ip	varchar	Representa la dirección IP asignada al estudio.
ip_rango	varchar	Representa el rango de IP asignado al estudio.

Tabla 8.Descripción de la tabla de base de datos: *study*

Nombre: <b>study_user_role</b>		
<b>Descripción</b>	Esta entidad generada contiene la relación entre los estudios usuarios y roles del sistema informático.	
Atributo	Tipo	Descripción
Id	bigint	Representa el identificador único de esta entidad, el cual es generado automáticamente.
role_name	varchar	Nombre del rol en el sistema.
study_id	numeric	Indica el identificador del estudio.
status_id	numeric	Indica el identificador de la entidad estado.
date_created	date	Representa la fecha de confección de la entidad study_user_role.

## *Implementación del módulo Configuración del Sistema de Gestión de Ensayos Clínicos*

---

date_updated	date	Representa la fecha de actualización de la entidad study_user_role.
user_name	varchar	Representa el nombre del usuario.
ip	varchar	Representa las direcciones IP de la entidad study_user_role.
ip_rango	varchar	Representa el rango de IP de la entidad study_user_role.

Tabla 9.Descripción de la tabla de base de datos: *study\_user\_role*

Nombre: study_parameter		
Descripción	Esta entidad contiene los parámetros introducidos por el usuario para llevar a cabo la configuración del estudio.	
Atributo	Tipo	Descripción
study_parameter_id	Bigint	Representa el identificador único de esta entidad, el cual es generado automáticamente.
Handle	Varchar	Nombre del parámetro recogido.
Name	Varchar	Representa la descripción del parámetro.
Description	Varchar	Indica la descripción del parámetro de configuración del estudio.
default_value	Varchar	Valor por defecto de la entidad.

Tabla 10.Descripción de la tabla de base de datos: *study\_parameter*

Nombre: study_parameter_value		
Descripción	La entidad contiene los parámetros introducidos por el usuario para la configuración del estudio y además sus valores correspondientes.	
Atributo	Tipo	Descripción
study_parameter_value_id	Bigint	Representa el identificador único de la entidad, el cual es generado automáticamente.
study_id	Integer	Representa el identificador del estudio.

# *Implementación del módulo Configuración del Sistema de Gestión de Ensayos Clínicos*

---

Value	Varchar	Indica el valor del parámetro introducido.
Parameter	Varchar	Indica el parámetro de configuración del estudio.

Tabla 11. Descripción de la tabla de base de datos: *study\_parameter\_value*

<b>Nombre: user_account</b>		
Descripción	En esta entidad se insertan los datos de los usuarios después de creados, también para mostrar el listado de usuarios, se cargan los datos a ilustrar de dicha tabla, así como otras funcionalidades con respecto a los usuarios.	
Atributo	Tipo	Descripción
user_id	Bigserial	Representa el identificador único del usuario, el cual es generado automáticamente.
user_name	Varchar	Nombre del usuario.
Passwd	Varchar	Representa la contraseña asignada al usuario.
first_name	Varchar	Indica el primer nombre del usuario.
last_name	Varchar	Indica los apellidos del usuario.
Email	Varchar	Contiene el correo electrónico del usuario.
active_study	Numeric	Representa el estudio activo al que pertenece el usuario.
institutional_affiliation	Varchar	Institución a la que pertenece el usuario.
status_id	Bigint	Indica el identificador de la entidad estado.
date_created	Date	Fecha de creación del usuario.
date_updated	Date	Fecha de actualización del usuario.
date_lastvisit	Date	Fecha de la última visita del usuario.
passwd_timestamp	Date	Período de duración de la contraseña asignada al usuario.

## *Implementación del módulo Configuración del Sistema de Gestión de Ensayos Clínicos*

---

passwd_challenge_question	Varchar	Representa la pregunta en caso de olvidada la contraseña de un usuario en específico.
passwd_challenge_answer	Varchar	Contiene la nueva contraseña introducida por el usuario.
Phone	Varchar	Indica el teléfono del usuario.
update_id	Numeric	Indica el identificador del rol que actualizo los datos de un usuario en específico.
Conectado	Integer	Representa si es usuario está habilitado o deshabilitado.
Ip	Varchar	Indica las direcciones IP asignadas al usuario.
ip_rango	Varchar	Indica el rango IP asignado al usuario.

Tabla 12.Descripción de la tabla de base de datos: *user\_account*

<b>Nombre: user_role</b>		
<b>Descripción</b>	La entidad generada contiene la relación entre los usuarios y roles de la aplicación.	
Atributo	Tipo	Descripción
role_id	Serial	Representa el identificador del rol asignado.
role_name	Varchar	Indica el nombre del rol asignado al usuario.
role_desc	Varchar	Descripción del rol asignado al usuario.

Tabla 13.Descripción de la tabla de base de datos: *user\_role*

<b>Nombre: status</b>		
<b>Descripción</b>	Esta entidad contiene los diferentes estados que tienen las entidades del sistema.	
Atributo	Tipo	Descripción
status_id	Bigserial	Representa el identificador del estado.

## *Implementación del módulo Configuración del Sistema de Gestión de Ensayos Clínicos*

---

Name	Varchar	Indica el nombre del estado.
Description	Varchar	Descripción del estado.

Tabla 14.Descripción de la tabla de base de datos: *status*

<b>Nombre: usuario</b>		
<b>Descripción</b>	En esta entidad se insertan los datos de los usuarios después de creados, también para mostrar el listado de usuarios, se cargan los datos a ilustrar de la misma, así como otras funcionalidades con respecto a los usuarios.	
Atributo	Tipo	Descripción
Id	Bigint	Representa el identificador único del usuario.
Nombre	Varchar	Nombre del usuario.
Password	Varchar	Representa la contraseña asignada al usuario.
fecha_nacimiento	Date	Indica la fecha de nacimiento del usuario.
Eliminado	Boolean	Representa con un valor (V o F) si es usuario está eliminado o no.
cid	Bigint	Carné de identidad del usuario.
cuenta_habilitada	Boolean	Representa con un valor (V o F) si es usuario habilitado o deshabilitado.
primer_apellido	Varchar	Indica el primer apellido del usuario.
segundo_apellido	Varchar	Indica el primer segundo del usuario.
direccion_particular	Varchar	Contiene la dirección particular del usuario.

## *Implementación del módulo Configuración del Sistema de Gestión de Ensayos Clínicos*

---

Telefono	Varchar	Indica el número de teléfono del usuario.
----------	---------	---

Tabla 15.Descripción de la tabla de base de datos: *usuario*

<b>Nombre: role</b>		
<b>Descripción</b>	Esta entidad contiene los roles creados en la aplicación y todos los datos referente a los mismos.	
Atributo	Tipo	Descripción
Name	Varchar	Representa el nombre del atributo creado en el sistema.
Eliminado	Boolean	Indica si un estudio fue eliminado o no (V o F) del sistema.
cid	Bigint	Contiene el carné de identidad del usuario.

Tabla 16.Descripción de la tabla de base de datos: *role*

<b>Nombre: user_in_role</b>		
<b>Descripción</b>	Esta entidad generada contiene la relación entre los usuarios y roles de la aplicación.	
Atributo	Tipo	Descripción
id_user	Bigint	Representa el identificador del usuario.
id_role	Bigint	Indica el identificador del rol asignado al usuario.

Tabla 17.Descripción de la tabla de base de datos: *user\_in\_role*

### **4.2 Modelo de Implementación**

El Modelo de Implementación es comprendido por un conjunto de componentes y subsistemas que constituyen la composición física de la implementación del software. Entre los componentes se pueden encontrar datos, archivos, ejecutables, código fuente y los directorios. Fundamentalmente, se describe la relación que existe desde los paquetes y clases del modelo de diseño a subsistemas y componentes

# Implementación del módulo Configuración del Sistema de Gestión de Ensayos Clínicos

físicos. El mismo describe cómo se implementan los componentes, agrupándolos en subsistemas organizados en capas y jerarquías, y señala las dependencias entre éstos. (42)

## 4.2.1 Diagrama de Despliegue

Es un diagrama estructurado que muestra la arquitectura del sistema desde el punto de vista del despliegue. Modela la arquitectura en tiempo de ejecución de un software. Esto muestra la configuración de los elementos de hardware (nodos) y muestra cómo los elementos y artefactos del sistema informático se trazan en esos nodos. (43)

Teniendo en cuenta las características de la aplicación a continuación se muestra como quedó estructurado el Diagrama de Despliegue:

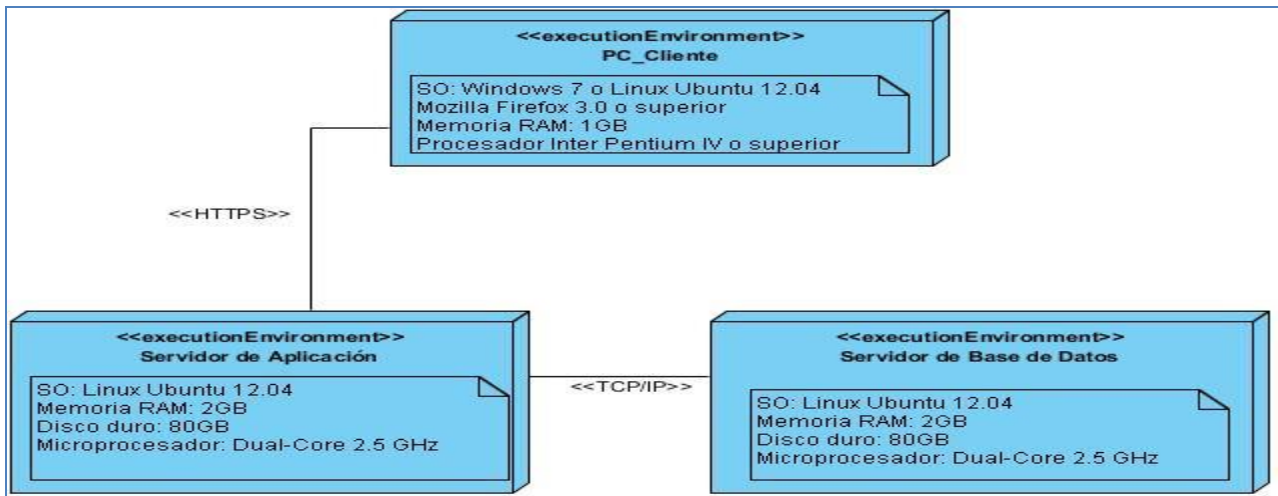


Fig. 8. Diagrama de Despliegue: módulo Configuración

## 4.2.2 Diagrama de Componentes

Un diagrama de componentes permite visualizar con más facilidad la estructura general del sistema y el comportamiento del servicio que estos componentes proporcionan y utilizan a través de las interfaces, se puede utilizar además para describir un diseño que se implemente en cualquier lenguaje o estilo. Solo es necesario identificar los elementos del diseño que interactúan con otros elementos del diseño a través de un conjunto restringido de entradas y salidas. Los componentes pueden tener cualquier escala y pueden estar interconectados de cualquier manera. (44)

# Implementación del módulo Configuración del Sistema de Gestión de Ensayos Clínicos

Teniendo en cuenta las características del módulo Configuración, a continuación se muestra como quedó estructurado el Diagrama de Componentes:

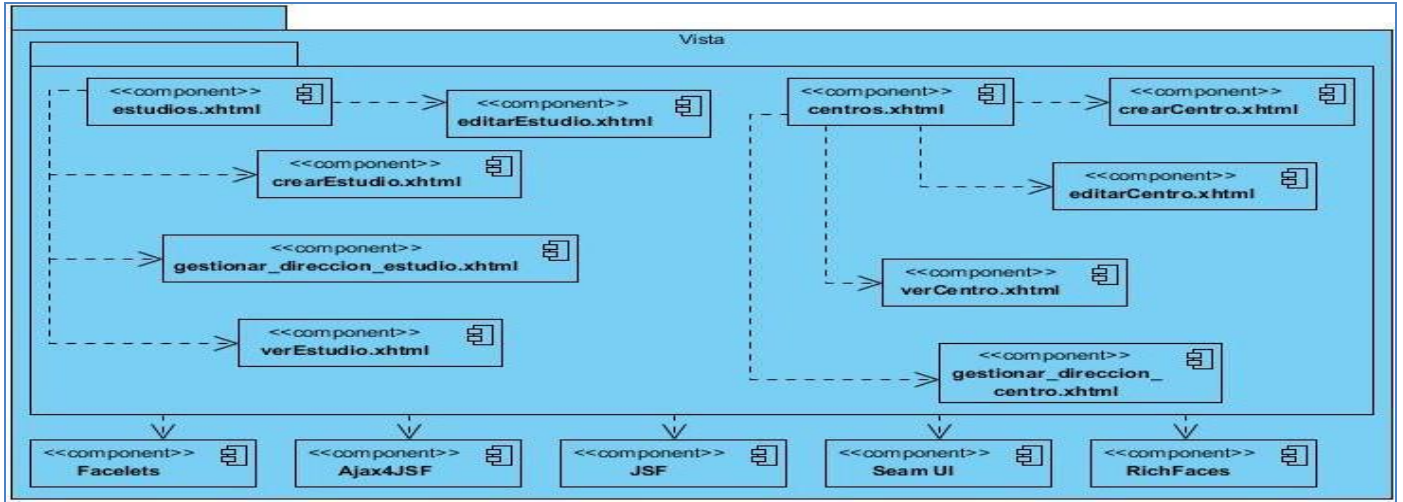


Fig. 9. Diagrama de Componentes: *Paquete Vista*

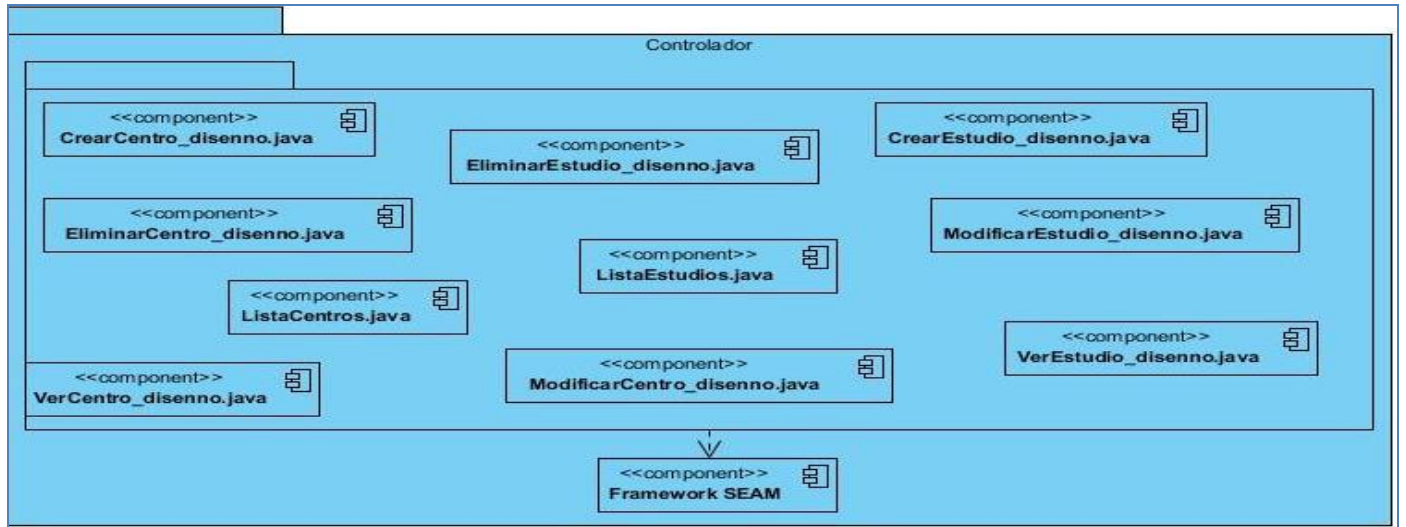


Fig. 10. Diagrama de Componentes: *Paquete Controlador*



# Implementación del módulo Configuración del Sistema de Gestión de Ensayos Clínicos

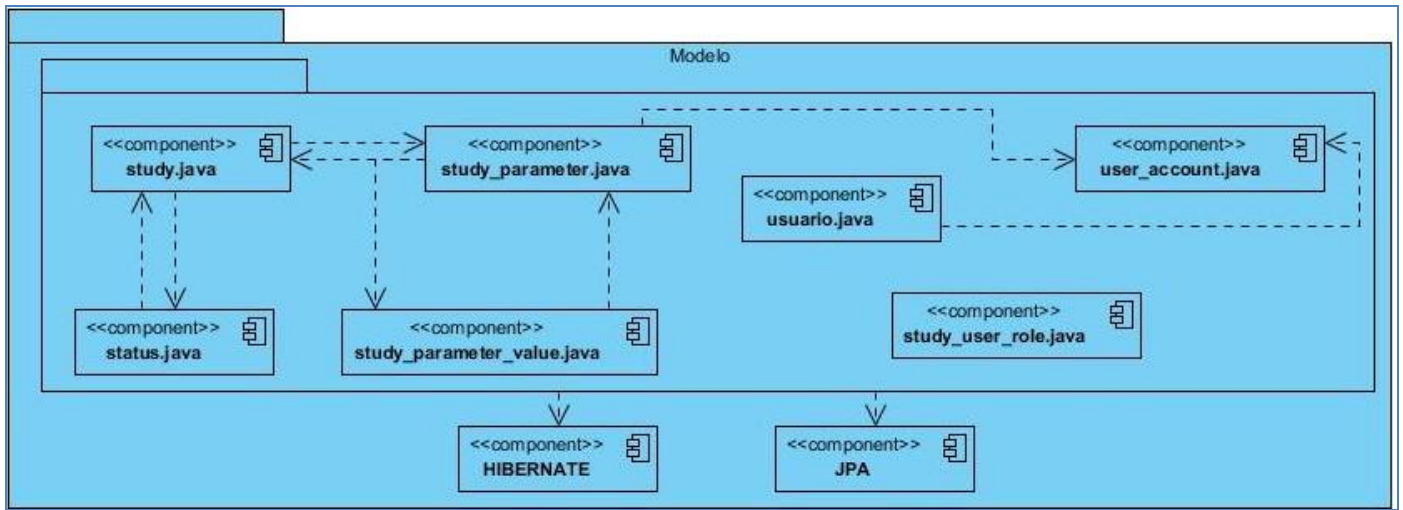


Fig. 11. Diagrama de Componentes: *Paquete Modelo*

## 4.3 Tratamiento de errores

La aparición de errores durante la ejecución de un programa resulta inevitable, por lo que un correcto tratamiento de ellos garantiza considerablemente un aumento en la calidad de las aplicaciones desarrolladas. La técnica de programación recomendada para erradicar la propagación de los errores producidos, es el tratamiento de excepciones o tratamiento de errores como también es conocido.

En las funcionalidades propuestas, para el tratamiento de excepciones, se hace uso de todos los elementos que brinda la plataforma, llevándose a cabo en toda porción de código donde puedan ocurrir situaciones inesperadas, principalmente en sentencias que manipulan datos de la base de datos.

El tratamiento de errores en las mismas, se realiza a partir del tratamiento de excepciones de persistencia de datos, así como la validación y conversión de los valores de entrada introducidos por el usuario, haciéndose posible mediante la aplicación del bloque **try catch** en porciones de código donde resulta necesario capturar y manejar determinadas excepciones.

Además el empleo de controles de conversión y validación facilitados por algunas de las tecnologías utilizadas en el desarrollo de estas funcionalidades; como el validador proporcionado por el framework Hibernate, en el que las restricciones y constantes de validación son expresadas mediante anotaciones en el Modelo (Entidades) y chequeadas en distintas capas de la aplicación. Por otra parte JBoss Seam y JSF permiten iniciar el proceso de validación desde la capa de presentación de la aplicación, en el que los

# *Implementación del módulo Configuración del Sistema de Gestión de Ensayos Clínicos*

---

mensajes correspondientes a cada una de las excepciones capturadas son presentados en la Vista, y las constantes en cambio son expresadas y procesadas en las capas Modelo y Controlador.

## **4.4 Seguridad**

Se considera un sistema informático seguro cuando garantiza la confidencialidad (la información sólo debe ser visible para las personas autorizadas), integridad (la información solo puede ser modificada por las personas autorizadas y de la forma autorizada) y disponibilidad (la información es accedida por las personas autorizadas en el momento requerido) de sus datos. El módulo Configuración del sistema SIGEC 1.0 provee una serie de funcionalidades que aseguran el subsistema, entre ellas se encuentra la autenticación del usuario.

Para acceder al sistema el usuario que trabajará con la aplicación deberá autenticarse introduciendo su usuario y contraseña correspondiente. El software por su parte verifica que los datos introducidos sean válidos y autoriza al mismo a los niveles establecidos de la aplicación según el rol asignado y permisos otorgados a este.

Por otra parte las contraseñas sólo pueden ser cambiadas por el propio usuario o por el administrador del sistema informático. Los datos ingresados en la aplicación no serán eliminados físicamente de la base de datos. Además se llevará un registro de trazas, el cual permitirá almacenar todas las acciones realizadas por el usuario.

## **4.5 Estrategias de codificación. Estándares y estilos a utilizar**

Un estándar de codificación comprende todos los aspectos de la generación de código. Si bien los programadores deben implementar un estándar de forma prudente, éste debe tender siempre a lo práctico. Estos establecen una serie de reglas que los programadores deben seguir para que el código sea entendible por cualquier desarrollador del grupo de trabajo; creando las condiciones para un futuro mantenimiento del sistema más rápido y eficiente, ya sea creando nuevas funcionalidades o modificando las ya existentes.

Para definir las estrategias de codificación a seguir en el desarrollo de la aplicación se utilizaron las notaciones estándar establecidas para sistemas informáticos realizados en Java, tales como:

# *Implementación del módulo Configuración del Sistema de Gestión de Ensayos Clínicos*

---

**Notación Pascal Casing:** consiste en que los identificadores y nombres de variables, métodos y funciones están compuestos por múltiples palabras unidas iniciando cada palabra con letra mayúscula. Ejemplo: UsuarioPromotor.

**Notación Camel:** consiste en escribir los identificadores con la primera letra de cada palabra en mayúsculas y el resto en minúscula.

Para la nomenclatura basada en el estándar a utilizar se muestran a continuación algunas excepciones:

## **Identación**

Inicio y bloque de fin: se recomienda dejar dos espacios en blanco desde la instrucción anterior para el inicio y fin de bloque ({}). Lo mismo sucede para el caso de las instrucciones if, else, for, while, do while, switch, foreach.

Aspectos generales: el indentado debe ser de dos espacios por bloque de código. No se debe usar el tabulador; ya que este puede variar según la configuración de dicha tecla. Los inicios ({} y cierres (}) de ámbito deben estar alineados debajo de la declaración a la que pertenecen y deben evitarse si hay sólo una instrucción.

## **Variables y constantes**

Para las variables y constantes el nombre asignado para ellas, debe permitir que con sólo leerlo se identifique el propósito de esta, así como también debe de empezar con la primera letra en minúscula e identificar además el tipo de dato de la misma. En caso de que sea un nombre compuesto, la segunda palabra, iniciará con letra mayúscula.

## **Comentarios, separadores, líneas, espacios en blanco y márgenes**

Comentarios: se recomienda escribirlos al inicio de cada clase o función de forma tal que estos describan el objetivo de la misma, así como los parámetros que usa (declarar tipos de datos, y objetivo del parámetro).

Líneas en blanco: se recomienda dejar una línea en blanco antes y después de la declaración de una clase o estructura y de la implementación de una función.

Espacios en blanco: se deben usar espacios en blanco entre operadores lógicos y aritméticos para lograr mayor entendimiento del código. Ejemplo: estudio = nombreEstudio. Además no se debe usar espacio en

# *Implementación del módulo Configuración del Sistema de Gestión de Ensayos Clínicos*

---

blanco después del corchete abierto y antes del cerrado de un arreglo, luego del paréntesis abierto y antes del cerrado o antes de un punto y coma.

## **Clases y objetos**

El nombre asignado a las clases deben comenzar en mayúscula y el resto de las letras en minúscula, en caso de ser un nombre compuesto se empleará notación Pascal Casing. Ejemplo: MiEstudio(). Para el caso de las instancias se comenzará con un prefijo que identificará el tipo de dato, el mismo se escribirá en minúscula.

Para los atributos de las clases, el nombre asignado debe empezar en minúscula y estará en correspondencia al tipo de dato al que se refiere, en caso de ser un nombre compuesto, la segunda palabra comenzará con mayúscula.

Como aspecto general el nombre empleado para las clases, objetos, atributos y funciones debe permitir que con sólo leerlo se identifique el propósito de estas.

La realización del presente capítulo proporcionó una visualización de las relaciones físicas entre los componentes de hardware y software del sistema informático, especificando los requisitos mínimos que estos deben cumplir. El desarrollo de un software robusto, flexible y con atributos sólidos de seguridad ha sido favorecido a través del manejo de aspectos como el tratamiento de excepciones, el establecimiento de la seguridad de la aplicación y las estrategias de codificación empleadas.

### **CONCLUSIONES**

Con el desarrollo de estas funcionalidades para el Sistema de Gestión de Ensayos Clínicos se ha cumplido con el objetivo general y con las tareas de la investigación definidas, obteniéndose las siguientes conclusiones:

- El estudio realizado de los sistemas existentes a nivel internacional, demostró que estos no resuelven la problemática identificada, pues solo son factibles en un estrecho dominio. A nivel nacional, aprovechando las características y ventajas que posee el HIS, se reutilizarán funcionalidades correspondientes al módulo Configuración del mismo, que permitirán disminuir el tiempo de desarrollo del software y posibilitará una futura integración con dicho sistema.
- El estudio y aplicación del Proceso Unificado de Desarrollo de Software, constituyó una buena práctica, que permitió la obtención exitosa de los artefactos generados durante el desarrollo de las funcionalidades propuestas.
- Se realizó el análisis y diseño del módulo Configuración y se implementaron las funcionalidades correspondientes a la gestión de estudios y centros apoyado por el análisis de las tecnologías, herramientas y pautas de diseño definidas por el Departamento de Sistemas Especializados en Salud para el desarrollo de sus aplicaciones, posibilitando a los especialistas el trabajo con la información válida.

## *Recomendaciones*

---

### **RECOMENDACIONES**

Una vez concluido el trabajo de diploma, el equipo de desarrollo recomienda:

- Implementar las funcionalidades correspondientes a la gestión de usuarios, roles y permisos del módulo Configuración con el objetivo de lograr la confidencialidad, integridad y disponibilidad de la información del Sistema de Gestión de Ensayos Clínicos 1.0.
- Agregar la funcionalidad que permita a los usuarios realizar la solicitud de cuenta al administrador. Esta funcionalidad posibilitará a los especialistas de otras instituciones y provincias solicitar una cuenta de usuario a través de la pantalla de inicio, disminuyendo el número de pasos requeridos para crear una cuenta de usuario nueva.

## Referencias Bibliográficas

---

### REFERENCIAS BIBLIOGRÁFICAS

1. **Espinoza, Cindia. Espacio MIS-TI y MATI.** Las Tecnologías de Información y Comunicación(TIC's) en salud. [En línea] 22 de 6 de 2010. [Citado el: 11 de 12 de 2013.] <http://mati2010.wordpress.com/2010/06/22/las-tecnologias-de-informacion-y-comunicacion-tic%E2%80%99s-en-salud/>.
2. **Sociedad Española Farmacológica Clínica .**Ensayos Clínicos . [En línea] Sociedad Española, 2013. [Citado el: 02 de 12 de 2013.] <http://se-fc.org/gestor/ensayos-clinicos/informacion-general.html>
3. **Perle. perle-Essential Network Connectivity.** Uso de seguridad AAA para la administración de Coversores de Medios y Extensores Ethernet [En línea] 2014. [Citado el: 25 de 04 de 2014.] <http://www.perlesystems.es/supportfiles/AAA-Security.shtml>.
4. **Gómez Velázquez, Karel.**PROPUESTA DE PROCESOS DE AUTENTICACIÓN, AUTORIZACIÓN Y AUDITORÍA PARA APLICACIONES BASADAS EN SERVICIOS WEB XML. La Habana, UCI : s.n., 2010.
5. **S4Research.** S4Research. acerca de Openclínica [En línea] 2012. [Citado el: 20 de 02 de 2014.] <http://www.s4research.es/openclinica.php>
6. **Megapractical . megapractical.**Soluciones Médicas Integrales [En línea] [Citado el: 01 de 03 de 2014.] <http://megapractical.com/index.php/producto-his>
7. **Martínez Pérez, Noslen.** Gestión de incidencias de recetas médicas en farmacia para el Módulo Control de Recetas Médicas. La Habana,Cuba : s.n., 2012.
8. **Allsoft. Allsoft.** El Modelo CMMI [En línea] 2013. [Citado el: 25 de 01 de 2014.] <http://www.allsoft.com.mx/recursos/EIModeloCMMI.pdf>.
9. **Díaz Flores, Miriam Milagros.** USMP-Universidad de San Martin de Probres. Proceso Unificado de Desarrollo de Software [En línea] 2014. [Citado el: 25 de 01 de 2014.] <http://www.usmp.edu.pe/publicaciones/boletin/fia/info49/articulos/RUP%20vs.%20XP.pdf>.
10. **Villares Arias, Javier y García Hernández, Yanisbel.**Consultas especializadas del servicio de Estomatología para el Sistema de Información Hospitalaria: alas HIS. La Habana : s.n., 2009.

## Referencias Bibliográficas

---

11. **Morales Izaguirre, Yamila y Cabrera Sanz, Victor Eduardo.** Componente para la propuesta de tratamiento a pacientes de la especialidad de oftalmología del sistema alas HIS. La Habana : s.n., 2013.
12. **Milián Pérez, Reinier y Rodríguez Sorí, René. Serie Científica-Universidad de las Ciencias Informáticas.** Monitorización en tiempo real de pacientes en un sistema de información hospitalaria [En línea] 2012. [Citado el: 25 de 01 de 2014.] <http://publicaciones.uci.cu/index.php/SC/article/view/985/566>.
13. **Nava Camacho, Cesar. Desarrollo de aplicaciones con Java.** ¿ Que son los JBoss Tools? [En línea] 2011. [Citado el: 01 de 02 de 2014.] <http://nvcesar.blogspot.com/2011/04/que-son-los-jboss-tools.html>.
14. **Manager, Free Download. Free Download Manager.** Visual Paradigm for UML. [En línea] 2008. [Citado el: 01 de 02 de 2014.] [http://www.freedownloadmanager.org/es/downloads/Paradigma\\_Visual\\_para\\_UML\\_%28M%C3%8D%29\\_14720\\_p/](http://www.freedownloadmanager.org/es/downloads/Paradigma_Visual_para_UML_%28M%C3%8D%29_14720_p/).
15. **S.A, Software y Consultoría Proemsa** Software de seguridad y auditoría [En línea] [Citado el: 01 de 04 de 2014.] <http://www.proemsa.com/administracion-de-seguridad-y-auditoria/>
16. **Scribd Inc. Scribd.** Investigación PostgreSQL [En línea] [Citado el: 02 de 02 de 2014.] <http://es.scribd.com/doc/36570462/postgreSQL-investigacion>.
17. **Ken, Arnold, Gosling, James y Holmes, David. Lenguajes de Programación .** Lenguaje de programación: Programación Java. [En línea] 2009. [Citado el: 05 de 02 de 2014.] <http://www.lenguajes-de-programacion.com/programacion-java.shtml>.
18. **Eguiluz, Javier. LIBROSWEB.** Introduccion a JavaScript [En línea] [Citado el: 20 de 02 de 2014.] [http://librosweb.es/javascript/capitulo\\_1.html](http://librosweb.es/javascript/capitulo_1.html).
19. **Ferguson, Jhon. WAKALEO CONSULTING.** JSF Jumpstart: Getting up to speed with Java Server Faces [En línea] 2009. [Citado el: 06 de 02 de 2014.] [http://www.wakaleo.com/public\\_resources/jsf-jumpstarter.pdf](http://www.wakaleo.com/public_resources/jsf-jumpstarter.pdf).
20. **decodigo. decodigo.com.** JavaServer Faces 2.0. Que es Facelets [En línea] 2010. [Citado el: 06 de 02 de 2014.] <http://www.decodigo.com/2011/08/que-es-facelets-52.html>.
21. **Martinez, Jose. slideshare.** slideshare,Introducción a Java Persistence API. [En línea] 2009. [Citado el: 11 de 02 de 2014.] <http://www.slideshare.net/jamslug/introduccion-a-java-persistence-api>.



## Referencias Bibliográficas

---

22. **Web Application. Plataforma J2EE.** Desarrollo de aplicaciones web bajo entornos Java, frameworks de desarrollo J2EE. JBoss Seam Framework. [En línea] febrero de 2009. [Citado el: 10 de 02 de 2014.] <http://wilmanchamba.wordpress.com/2008/02/20/jboss-seam-framework/>.
23. **Community, JBoss.** HIBERNATE: Hibernate Tools. [En línea] 2010. [Citado el: 10 de 02 de 2014.] <http://hibernate.org/subprojects/tools.html>.
24. **Computación, Departamento de Ingeniería de Sistemas y. Universidad de los Andes.** Universidad de los andes, Departamento de sistemas: Enterprise Java Bean 3 (EJB 3), Arquitectura de software [En línea] 2012. [Citado el: 10 de 02 de 2014.] <http://sistemas.uniandes.edu.co/~isis3702/dokuwiki/lib/exe/fetch.php?media=principal:isis3702-ejb3.pdf>.
25. **Canales Mora, Roberto. Adictos al Trabajo.** Tutoriales: Introducción a Ajax4Jsf. [En línea] 2008. [Citado el: 06 de 02 de 2014.] <http://www.adictosaltrabajo.com/tutoriales/tutoriales.php?pagina=Ajax4Jsf>.
26. **Adictos al Trabajo.** Tutoriales: Introducción a RichFaces. [En línea] 2010. [Citado el: 08 de 02 de 2014.] <http://www.adictosaltrabajo.com/tutoriales/tutoriales.php?pagina=richFacesJsflIntro>.
27. **Corbillón, Javier Olover.** Desarrollo de funcionalidades del módulo Facturación del sistema alas HIS orientado al consumo de productos farmacéuticos. La Habana, Cuba : s.n., 2012.
28. **Bill, Shannon. oracle.com.** Oracle Technology Network: Java EE 5 Technologies. [En línea] 2010. [Citado el: 12 de 02 de 2014.] <http://www.oracle.com/technetwork/java/javaee/tech/javaee5-jsp-135162.html>.
29. **Genos. Genos.** Soporte JBoss. [En línea] 2013. [Citado el: 15 de 02 de 2014.] <http://www.genos.es/soporte/jboss-soporte>.
30. **Venners, Bill. JavaWorld.** The lean, mean, virtual machine. An introduction to the basic structure and functionality of the Java Virtual Machine. [En línea] 2014. [Citado el: 13 de 02 de 2014.] <http://www.javaworld.com/article/2077184/core-java/the-lean--mean--virtual-machine.html>.
31. **UPM, Facultad de Informática -. Unidad Docente de Ingeniería del Software (UDIS).** Modelo de Dominio [En línea] [Citado el: 05 de 03 de 2014.] <http://is.ls.fi.upm.es/docencia/is2/documentacion/ModeloDominio.pdf>.
32. **WordPress.com. Tecnología y Sinergex.** Requisitos funcionales y No funcionales [En línea] 2010. [Citado el: 05 de 03 de 2014.] <http://synergix.wordpress.com/2008/07/07/requisito-funcional-y-no-funcional/>.

## Referencias Bibliográficas

---

33. **Quiroga, Juan Pablo. Universisda de los Andes - Colombia.** Requerimientos [En línea] 2012. [Citado el: 05 de 03 de 2014.] <http://sistemas.uniandes.edu.co/~csof5101/dokuwiki/lib/exe/fetch.php?media=principal:csof5101-requerimientos.pdf>.
34. **Synergix. Tecnología y Synergix.** Tipos de requisitos: Funcional vs. No Funcional [En línea] 2009. [Citado el: 08 de 03 de 2014.] <http://synergix.wordpress.com/2008/07/25/definimos-actor-como/>.
35. **Navarro, Juan José Moreno. babel-Universidad Poletitécnica de Madrid.** Arquitecturas Software [En línea] [Citado el: 10 de 03 de 2014.] [http://babel.ls.fi.upm.es/~fred/sbc/arquitecturas\\_sw.pdf](http://babel.ls.fi.upm.es/~fred/sbc/arquitecturas_sw.pdf).
36. **Pavón Mestras, Juan. Universidad Complutense-Informática.** Estructura de las Aplicaciones Orientadas a Objetos. El patrón Modelo-Vista-Controlador (MVC) [En línea] 2009. [Citado el: 04 de 02 de 2014.] <https://www.fdi.ucm.es/profesor/jpavon/poo/2.14.MVC.pdf>.
37. **Peinado, Federico. Universidad de Complutense-Informática.** LPS: Patrón. Modelo-Vista-Controlador. Facultad de Informática. [En línea] 2009. [Citado el: 03 de 02 de 2014.] <http://web.fdi.ucm.es/profesor/fpeinado/courses/oop/LPS-14ModeloVistaControlador.pdf>.
38. **Nuñez, Cheko. scribd.** Las Estrategias de Integración [En línea] 3 de 05 de 2011. [Citado el: 10 de 03 de 2014.] <http://es.scribd.com/doc/54480592/Las-Estrategias-de-Integracion>.
39. **Grosso, Andrés. Prácticas de Software.** Patrones GRASP [En línea] 21 de marzo de 2011. [Citado el: 10 de 04 de 2014.] <http://www.practicadesoftware.com.ar/2011/03/patrones-grasp/>.
40. **Bustos, Juan Pablo. slideshare.** Diagramas de Clases [En línea] 09 de Agosto de 2011. [Citado el: 12 de 03 de 2014.] <http://www.slideshare.net/jpbthames/diagramas-de-clases>.
41. **Fletes Gudiño, Pedro y Hernández Barbosa, Oscar Daniel. Instituto Tecnológico de Colima- Departamento de Sistemas y Computación.** Tutorial de Fundamentos de bases de datos: Definición de modelo de datos [En línea] 2009. [Citado el: 16 de 04 de 2014.] [http://labredes.itcolima.edu.mx/fundamentosbd/sd\\_u2\\_1.htm](http://labredes.itcolima.edu.mx/fundamentosbd/sd_u2_1.htm).
42. **Marrero, Carlos David, y otros. MeRinde.** Artefactos-Modelo de Implementación .[En línea] 2013. [Citado el: 16 de 04 de 2014.] [http://merinde.net/index.php?option=com\\_content&task=view&id=495&Itemid=291](http://merinde.net/index.php?option=com_content&task=view&id=495&Itemid=291).
43. **SOLUS S.A. Sparx System Argentina.** Tutorial UML 2.0- Diagrama de Despliegue UML 2 [En línea] 2009. [Citado el: 16 de 04 de 2014.] [http://www.sparxsystems.com.ar/resources/tutorial/uml2\\_deploymentdiagram.html](http://www.sparxsystems.com.ar/resources/tutorial/uml2_deploymentdiagram.html).

## *Referencias Bibliográficas*

---

44. **Microsoft. Microsoft - Developer Network.** Diagramas de componentes de UML [En línea] 06 de 14 de 2013. [Citado el: 18 de 04 de 2014.] <http://msdn.microsoft.com/es-es/library/dd409390.aspx>.
45. **Herrera Pérez, Leonelbys y Hernández Cuello, Leandro Evangelio.** alasClínica. Desarrollo de las funcionalidades del Cronograma de Ejecución de los Ensayos Clínicos Cubanos en el sistema OpenClinica. La Habana : s.n., 2009.
46. **Rodríguez, Lucia, y otros.**“ALASCLÍNICAS”: SISTEMA DE GESTIÓN DE ENSAYOS CLÍNICOS. La Habana : s.n., 2011.
47. **Oracle. Oracle.com.** Oracle Applications - Health Sciences Solution - E-Clinical Solutions- Oracle Clinical [En línea] 2013. [Citado el: 22 de 01 de 2014.] <http://www.oracle.com/us/products/applications/health-sciences/e-clinical/clinical/index.html?ssSourceSiteId=opn>.
48. **Corporation, PAREXEL International. PAREXEL.** Parexel-Perceptive MyTrials [En línea] 2014. [Citado el: 24 de 01 de 2014.] <http://www.perceptive.com/mytrials/>.
49. **Villares Arias, Javier y García Hernández, Yanisbel.** Consultas especializadas del servicio de Estomatología para el Sistema de Información Hospitalaria: alas HIS. La Habana : s.n., 2009.
50. **SCM -Española. SCM.** SCM Soporte Intercontinental. Software Sanitario: ALAS HIS. [En línea] 2010. [Citado el: 25 de 01 de 2014.] <http://www.scm.es/scmsi/index.php/software-sanitario/alas-his>.
51. **Open ERP. Open ERP.** iReport. [En línea] 2014 [Citado el: 02 de 02 de 2014.] <http://openerpspain.com/ireport/>.
52. **Hernández Ramírez, Martha.** MANUAL DE USUARIO V2.0-alasClínicas. UCI, La Habana : s.n., 2011.

## BIBLIOGRAFÍA

1. **Adictos al Trabajo.** Adictos al Trabajo. Tutoriales: Introducción a RichFaces [En línea] 2010. [Citado el: 08 de 02 de 2014.] <http://www.adictosaltrabajo.com/tutoriales/tutoriales.php?pagina=richFacesJsflIntro>.
2. **Alvarez Lorenzo, Amaya y Mora Maure, Morelio.** Desarrollo de funcionalidades para la especialidad de Psicología del módulo Consulta Externa del sistema alas HIS. La Habana, Cuba : s.n., 2012.
3. **Allsoft. Allsoft.** El Modelo CMMI [En línea] 2013. [Citado el: 25 de 01 de 2014.] <http://www.allsoft.com.mx/recursos/EIModeloCMMI.pdf>.
4. **Bustos, Juan Pablo. slideshare.** Diagramas de Clases [En línea] 09 de Agosto de 2011. [Citado el: 12 de 03 de 2014.] <http://www.slideshare.net/jpbthames/diagramas-de-clases>.
5. **Canales Mora, Roberto. Adictos al Trabajo.** Tutoriales: Introducción a Ajax4Jsf. [En línea] 2008. [Citado el: 06 de 02 de 2014.] <http://www.adictosaltrabajo.com/tutoriales/tutoriales.php?pagina=Ajax4Jsf>.
6. **Community, JBoss. JBoss Community. Hibernate HIBERNATE: Hibernate Tools.** [En línea] 2010. [Citado el: 10 de 02 de 2014.] <http://hibernate.org/subprojects/tools.html>.
7. **Computación, Departamento de Ingeniería de Sistemas y. Universidad de los Andes.** Universidad de los andes, Departamento de sistemas: Enterprise Java Bean 3 (EJB 3), Arquitectura de software [En línea] 2012. [Citado el: 10 de 02 de 2014.] <http://sistemas.uniandes.edu.co/~isis3702/dokuwiki/lib/exe/fetch.php?media=principal:isis3702-ejb3.pdf>.
8. **Corbillón, Javier Olover.** Desarrollo de funcionalidades del módulo Facturación del sistema alas HIS orientado al consumo de productos farmacéuticos. La Habana, Cuba : s.n., 2012.
9. **Corporation, PAREXEL International. PAREXEL.** Parexel-Perceptive MyTrials [En línea] 2014. [Citado el: 24 de 01 de 2014.] <http://www.perceptive.com/mytrials/>.
10. **decodigo. decodigo.com.** JavaServer Faces 2.0. Que es Facelets [En línea] 2010. [Citado el: 06 de 02 de 2014.] <http://www.decodigo.com/2011/08/que-es-facelets-52.html>
11. **Díaz Flores, Miriam Milagros. USMP-Universidad de San Martín de Porres.** Proceso Unificado de Desarrollo [En línea] 2014. [Citado el: 25 de 01 de 2014.] <http://www.usmp.edu.pe/publicaciones/boletin/fia/info49/articulos/RUP%20vs.%20XP.pdf>.
12. **Eguiluz, Javier. LIBROSWEB.** Introducción a JavaScript [En línea] [Citado el: 20 de 02 de 2014.] [http://librosweb.es/javascript/capitulo\\_1.html](http://librosweb.es/javascript/capitulo_1.html).

## Bibliografía

---

13. **Espinoza, Cindia. Espacio MIS-TI y MATI.** Las Tecnologías de Información y Comunicación(TIC's) en salud. [En línea] 22 de 6 de 2010. [Citado el: 11 de 12 de 2013.] <http://mati2010.wordpress.com/2010/06/22/las-tecnologias-de-informacion-y-comunicacion-tic%E2%80%99s-en-salud/>.
14. **Ferguson, Jhon.** WAKALEO CONSULTING. JSF Jumpstart: Getting up to speed with Java Server Faces [En línea] 2009. [Citado el: 06 de 02 de 2014.] [http://www.wakaleo.com/public\\_resources/jsf-jumpstarter.pdf](http://www.wakaleo.com/public_resources/jsf-jumpstarter.pdf).
15. **Genos. Genos.** Soporte JBoss. [En línea] 2013. [Citado el: 15 de 02 de 2014.] <http://www.genos.es/soporte/jboss-soporte>.
16. **Gómez Velázquez, Karel.** PROPUESTA DE PROCESOS DE AUTENTICACIÓN, AUTORIZACIÓN Y AUDITORÍA PARA APLICACIONES BASADAS EN SERVICIOS WEB XML. La Habana, UCI : s.n., 2010.
17. **Grosso, Andrés. Prácticas de Software.** Patrones GRASP [En línea] 21 de marzo de 2011. [Citado el: 10 de 04 de 2014.] <http://www.practicadesoftware.com.ar/2011/03/patrones-grasp/>.
18. **Gutierrez, Demián. code compiling.** Lenguaje Unificado de Modelado [En línea] mayo de 2011. [Citado el: 16 de 03 de 2014.] [http://www.codecompiling.net/files/slides/UML\\_clase\\_06\\_UML\\_secuencia.pdf](http://www.codecompiling.net/files/slides/UML_clase_06_UML_secuencia.pdf).
19. **Hernández Méndez, Yoan** Diseño e implementación de un módulo de seguridad para el sistema de negocio EBMS DSerp UCI, La Habana. s.n., 2013
20. **Hernández Ramírez, Martha.** MANUAL DE USUARIO V2.0-alasClínicas. UCI,La Habana : s.n., 2011.
21. **Hernández Ramírez, Martha y Rodríguez García, Lucía.** ESPECIFICACIÓN DE REQUISITOS DE SOFTWARE V2.0-alasClinicas. La Habana, Cuba : s.n., 2010.
22. **Herrera Pérez, Leonelbys y Hernández Cuello, Leandro Evangelio.** alasClínica. Desarrollo de las funcionalidades del Cronograma de Ejecución de los Ensayos Clínicos Cubanos en el sistema OpenClinica. La Habana : s.n., 2009.
23. **InterContinental, SCM-Soporte. SCM-Soporte InterContinental.** Alas PACS. [En línea] [Citado el: 01 de 03 de 2014.] <http://www.scmsi.es/scmsi/images/pdf/alas-pacs.pdf>.
24. **Ken, Arnold, Gosling, James y Holmes, David.** Lenguajes de Programación. Lenguaje de programación: Programación Java.. [En línea] 2009. [Citado el: 05 de 02 de 2014.] <http://www.lenguajes-de-programacion.com/programacion-java.shtml>.

# Bibliografía

---

25. **Manager, Free Download.** Free Download Manager. Visual Paradigm [En línea] 2008. [Citado el: 01 de 02 de 2014.] [http://www.freedownloadmanager.org/es/downloads/Paradigma\\_Visual\\_para\\_UML\\_%28M%C3%8D%29\\_14720\\_p/](http://www.freedownloadmanager.org/es/downloads/Paradigma_Visual_para_UML_%28M%C3%8D%29_14720_p/).
26. **Marrero, Carlos David, y otros. MeRinde.** Artefactos-Modelo de Implementación [En línea] 2013. [Citado el: 16 de 04 de 2014.] [http://merinde.net/index.php?option=com\\_content&task=view&id=495&Itemid=291](http://merinde.net/index.php?option=com_content&task=view&id=495&Itemid=291).
27. **Martínez Jera, Erislan y González Enríquez, Landy.** Desarrollo de la Gestión de Datos de Ensayos Clínicos a partir del sistema OpenClinica. La Habana, Cuba : s.n., 2009.
28. **Martinez, Jose. Slideshare.** slideshare, Introducción a Java Persistence API. [En línea] 2009. [Citado el: 11 de 02 de 2014.] <http://www.slideshare.net/jamslug/introduccion-a-java-persistence-api>.
29. **Megapractical . megapractical.** Soluciones Médicas Integrales [En línea] [Citado el: 01 de 03 de 2014.] <http://megapractical.com/index.php/his-micro>.
30. **Microsoft. Microsoft - Developer Network.** Diagramas de componentes de UML [En línea] 06 de 14 de 2013. [Citado el: 18 de 04 de 2014.] <http://msdn.microsoft.com/es-es/library/dd409390.aspx>.
31. **Milián Pérez, Reinier y Rodríguez Sorí, René. Serie Científica-Universidad de las Ciencias Informáticas.** Monitorización en tiempo real de pacientes en un sistema de información hospitalaria [En línea] 2012. [Citado el: 25 de 01 de 2014.] <http://publicaciones.uci.cu/index.php/SC/article/view/985/566>.
32. **Mirabal, Ing. Lisseth Agüero. slideshare.** Requerimientos funcionales y No funcionales. [En línea] [Citado el: 05 de 03 de 2014.] <http://www.slideshare.net/Lismirabal/requerimientos-funcionales-y-no-funcionales>.
33. **Morales Izaguirre, Yamila y Cabrera Sanz, Víctor Eduardo.** Componente para la propuesta de tratamiento a pacientes de la especialidad de oftalmología del sistema alas HIS. La Habana : s.n., 2013.
34. **Nava Camacho, Cesar.** Desarrollo de aplicaciones con Java. Desarrollo de aplicaciones con Java. [En línea] 2011. [Citado el: 01 de 02 de 2014.] <http://nvcesar.blogspot.com/2011/04/que-son-los-jboss-tools.html>.
35. **Navarro, Juan José Moreno. babel-Universidad Politécnica de Madrid.** Arquitecturas Software [En línea] [Citado el: 10 de 03 de 2014.] [http://babel.ls.fi.upm.es/~fred/sbc/arquitecturas\\_sw.pdf](http://babel.ls.fi.upm.es/~fred/sbc/arquitecturas_sw.pdf).

## Bibliografía

---

36. **Neuland Agüero, Denis, y otros.** Programa de Mejora- 0523 Políticas. La Habana, UCI : s.n., 2011.
37. **Nuñez, Cheko. scribd.** Estrategia de Integración [En línea] 3 de 05 de 2011. [Citado el: 10 de 03 de 2014.] <http://es.scribd.com/doc/54480592/Las-Estrategias-de-Integracion>.
38. **Open ERP.** Open ERP. iReport. [En línea] [Citado el: 02 de 02 de 2014.] <http://openerpspain.com/ireport/>.
39. **Oracle. Oracle.com.** Oracle Applications - Health Sciences Solution - E-Clinical Solutions- Oracle Clinical [En línea] 2013. [Citado el: 22 de 01 de 2014.] <http://www.oracle.com/us/products/applications/health-sciences/e-clinical/clinical/index.html?ssSourceSiteId=opn>.
40. **Pavón Mestras, Juan. Universidad Complutense-Informática.** Patrón: Modelo Vista Controlador [En línea] 2009. [Citado el: 04 de 02 de 2014.] <https://www.fdi.ucm.es/profesor/jpavon/poo/2.14.MVC.pdf>.
41. **Peinado, Federico. Universidad de Complutense-Informática.** LPS: Patrón. Modelo-Vista-Controlador. Facultad de Informática. [En línea] 2009. [Citado el: 03 de 02 de 2014.] <http://web.fdi.ucm.es/profesor/fpeinado/courses/oop/LPS-14ModeloVistaControlador.pdf>.
42. **Peña García, Zahily y Soler Izquierdo, Geidar.** Desarrollo del componente de predicción de consumo de productos para el Sistema de Información Hospitalaria alas HIS. La Habana, UCI : s.n., 2013.
43. **Perle. perle-Essential Network Connectivity.** Uso de seguridad AAA para la administración de Coversores de Medios y Extensores Ethernet [En línea] 2014. [Citado el: 25 de 04 de 2014.] <http://www.perlesystems.es/supportfiles/AAA-Security.shtml>.
44. **Quiroga, Juan Pablo. Universisda de los Andes - Colombia.** Requerimientos [En línea] 2012. [Citado el: 05 de 03 de 2014.] <http://sistemas.uniandes.edu.co/~csof5101/dokuwiki/lib/exe/fetch.php?media=principal:csof5101-requerimientos.pdf>.
45. **Reyes Perdomo, Dailin y Moreno de León, Juan Miguel.** Desarrollo de los procesos para la gestión de la información del trasplante de riñón para el Sistema de Trasplante. La Habana, Cuba : s.n., 2013.
46. **Rodríguez Gómez, Hilda Maria y Moreira Rodríguez, Jorge.** Desarrollo del Módulo de Gestión de Reuniones del Sistema de Teleconsulta. La Habana, UCI : s.n., 2011.
47. **Rodriguez, Lucia, y otros.** "ALASCLÍNICAS": SISTEMA DE GESTIÓN DE ENSAYOS CLÍNICOS. La Habana : s.n., 2011.



## Bibliografía

---

48. **Sánchez Santos, Arlety y Pacheco Correa, Yuniór.** Mecanismo de control de acceso en los servicios web y gestión de trazas en el Sistema Nomenclador de Información. La Habana, UCI : s.n., 2013.
49. **SCM -Española. SCM.** SCM Soporte Intercontinental. Software Sanitario: ALAS HIS. [En línea] 2010. [Citado el: 25 de 01 de 2014.] <http://www.scmsi.es/scmsi/index.php/software-sanitario/alas-his>.
50. **S.A, Software y Consultoría Proemsa 2014 Software y Consultoría Proemsa.** Software de seguridad y auditoría [En línea] 2014. [Citado el: 01 de 04 de 2014.] <http://www.proemsa.com/administracion-de-seguridad-y-auditoria/>
51. **Scribd Inc. Scribd.** Investigación PostgreSQL [En línea] [Citado el: 02 de 02 de 2014.] <http://es.scribd.com/doc/36570462/postgresql-investigacion>.
52. **SOLUS S.A. Sparx System Argentina.** Diagrama UML [En línea] 2009. [Citado el: 16 de 04 de 2014.] [http://www.sparxsystems.com.ar/resources/tutorial/uml2\\_deploymentdiagram.html](http://www.sparxsystems.com.ar/resources/tutorial/uml2_deploymentdiagram.html).
53. **Sociedad Española Farmacológica Clínica .** Sociedad Española Farmacológica Clínica .Ensayos Clínicos- OpenClínica [En línea] Sociedad Española, 2013. [Citado el: 02 de 12 de 2013.] <http://sefc.org/gestor/ensayos-clinicos/informacion-general.html>.
54. **S4Research.** S4Research. Servicios-Soporte [En línea] 2012. [Citado el: 20 de 02 de 2014.] <http://www.s4research.es/servicios.php>.
55. **S4Research.** S4Research. acerca de Openclínica [En línea] 2012. [Citado el: 20 de 01 de 2014.] <http://www.s4research.es/openclinica.php>.
56. **Synergix. Tecnología y Synergix.** Definios de actor [En línea] 2009. [Citado el: 08 de 03 de 2014.] <http://synergix.wordpress.com/2008/07/25/definimos-actor-como/>.
57. **UPM, Facultad de Informática -. Unidad Docente de Ingeniería del Software (UDIS).** Modelo de Dominio [En línea] [Citado el: 05 de 03 de 2014.] <http://is.ls.fi.upm.es/docencia/is2/documentacion/ModeloDominio.pdf>.
58. **Venners, Bill. JavaWorld.** Maquina Vistrual [En línea] 2014. [Citado el: 13 de 02 de 2014.] <http://www.javaworld.com/article/2077184/core-java/the-lean--mean--virtual-machine.html>.
59. **Villares Arias, Javier y García Hernández, Yanisbel.** Consultas especializadas del servicio de Estomatología para el Sistema de Información Hospitalaria: alas HIS. La Habana : s.n., 2009.
60. **Web Application.Plataforma J2EE.** Desarrollo de aplicaciones web bajo entornos Java, frameworks de desarrollo J2EE. JBoss Seam Framework. [En línea] febrero de 2009. [Citado el: 10 de 02 de 2014.] <http://wilmanchamba.wordpress.com/2008/02/20/jboss-seam-framework/>.



## *Bibliografía*

---

61. **WordPress.com. Tecnología y Sinergex.** Requisitos funcionales y No funcionales [En línea] 2010. [Citado el: 05 de 03 de 2014.] <http://synergix.wordpress.com/2008/07/07/requisito-funcional-y-no-funcional/>.

## ANEXO 1

### Entrevista

#### **Pregunta 1:**

¿Qué problemas presenta el módulo Administrar Sistema del sistema Clínicas 1.0?

#### **Respuesta:**

Los problemas que presenta el módulo Administrar sistema del software Clínicas 1.0 se encuentran en la gestión de usuarios, estudios y centros, ya que no permiten ser eliminadas en el momento necesario, lo que trae consigo que se le esté mostrando información innecesaria al especialista que está trabajando con la aplicación, la cual puede traerle confusión. Además los roles definidos en este sistema son estáticos, acarreado como consecuencia que cada vez que se quiera crear un rol nuevo haya que reprogramar esta parte del sistema y que sea imposible asignarle o quitarle responsabilidades a los roles existentes.

#### **Pregunta 2:**

¿Cuáles son los principales problemas a los que se enfrentan los especialistas con la utilización del módulo Administrar Sistema del software Clínicas 1.0?

#### **Respuesta:**

El sistema al no permitir eliminar usuarios, centros y estudios, le muestra información prescindible al usuario, la cual puede crear confusión al mismo. Por otra parte que los roles estén definidos estáticos, puede traer como consecuencia que si el especialista necesita un nuevo rol, este no puede interactuar con el sistema, producto de su inexistencia, afectando significativamente la adaptabilidad del software a las necesidades del cliente.

#### **Pregunta 3:**

¿Por qué las nuevas funcionalidades identificadas para dar solución a la problemática en cuestión, no se integrarán al sistema Clínicas 1.0?

#### **Respuesta:**

Las funcionalidades identificadas no se integrarán al sistema Clínicas 1.0, pues en el desarrollo de dicha aplicación no se utilizó un framework, por lo que el programador tenía que realizar las conexiones a la

base de datos mediante código y el servidor de aplicaciones Apache Tomcat 5.5 utilizado es obsoleto en comparación con la versión actual 7.0, por lo que el soporte de este sistema informático sería sumamente costoso.

**Pregunta 4:**

¿Cuáles son los roles que desempeñan los especialistas del CIM en la ejecución de un ensayo clínico?

**Respuesta:**

Los roles que desempeñan los especialistas en la conducción de un ensayo clínico son: promotores, coordinadores, investigadores, monitores y gerente de datos.

**Pregunta 5:**

¿Cuando se ejecuta la funcionalidad eliminar usuario, centro, estudio o rol; estos son eliminados completamente de la base de datos?

**Respuesta:**

Esta información no se elimina de la base de datos, el usuario no la visualiza en la vista pero físicamente se encuentra en el sistema con el campo eliminado en verdadero.

Descripción de casos de uso

CU4: Ver detalles de centro	
<b>Objetivo</b>	Muestra al usuario los datos referentes a un centro existente en la aplicación.
<b>Actores</b>	Usuario

<b>Resumen</b>	El caso de uso inicia cuando el Usuario selecciona la opción Gestionar centros, el sistema muestra los centros existentes y las diferentes acciones asociadas a los mismos, el usuario selecciona la acción Ver detalles de centro y el sistema ilustra los datos asociados al centro especificado, termina el caso de uso.
<b>Referencias</b>	RF25: Ver detalles de centro. RF26: Listar centros. RF27: Buscar centro.
<b>Flujo Normal de los eventos</b>	
<b>Actor</b>	<b>Sistema</b>
1. Selecciona la opción "Gestionar estudios".	2. Muestra un listado de los estudios existentes en el sistema y las opciones: Ver, Modificar, Adicionar, Eliminar, Buscar, Habilitar, Deshabilitar, Gestionar direcciones IPestudiosy Gestionar centros.
3. Selecciona la opción "Gestionar centros".	4. Muestra un listado de los centros existentes en el sistema y las opciones: Ver, Modificar, Adicionar, Eliminar, Buscar, Habilitar, Deshabilitar y Gestionar direcciones IP centros.

5. Selecciona la acción Ver detalles de centro.	6. Muestra todos los datos referentes al centro especificado.
	7. El caso de uso termina.
<b>Puntos de extensión</b>	CU. Buscar centro

Tabla 18.Descripción del caso de uso: *Ver detalles de centro.*

<b>CU5: Eliminar centro</b>	
<b>Objetivo</b>	Elimina a un centro existente en la aplicación.
<b>Actores</b>	Usuario
<b>Resumen</b>	El caso de uso inicia cuando el Usuario selecciona la opción Gestionar centros, el sistema muestra los centros existentes y las diferentes acciones asociadas a los mismos, el usuario selecciona la acción Eliminar centro y el sistema elimina el centro especificado, termina el caso de uso.
<b>Referencias</b>	RF28: Eliminar centro. RF26: Listar centros. RF27: Buscar centro.
<b>Flujo Normal de los eventos</b>	

Actor	Sistema
1. Selecciona la opción "Gestionar estudios".	2. Muestra un listado de los estudios existentes en el sistema y las opciones: Ver, Modificar, Adicionar, Eliminar, Buscar, Habilitar, Deshabilitar, Gestionar direcciones IP estudios Gestionar centros.
3. Selecciona la opción "Gestionar centros".	4. Muestra un listado de los centros existentes en el sistema y las opciones: Ver, Modificar, Adicionar, Eliminar, Buscar, Habilitar, Deshabilitar y Gestionar direcciones IP centros.
5. Selecciona la acción Eliminar centro.	6. Elimina el centro especificado.
	7. El caso de uso termina.
<b>Puntos de extensión</b>	CU. Buscar centro

Tabla 19. Descripción del caso de uso: *Eliminar centro*.

<b>CU6: Habilitar centro</b>	
<b>Objetivo</b>	Activa un centro existente en la aplicación, luego de haber estado desactivado.

## Anexos

---

<b>Actores</b>	Usuario
<b>Resumen</b>	El caso de uso inicia cuando el Usuario selecciona la opción Gestionar centros, el sistema muestra los centros existentes y las diferentes acciones asociadas a los mismos, el usuario selecciona la acción Habilitar centro y el sistema activa el centro especificado y a su vez todas las funcionalidades correspondientes al mismo, termina el caso de uso.
<b>Referencias</b>	RF29: Habilitar centro. RF27: Buscar centro. RF26: Listar centros.

### Flujo Normal de los eventos

<b>Actor</b>	<b>Sistema</b>
1. Selecciona la opción "Gestionar estudios".	2. Muestra un listado de los estudios existentes en el sistema y las opciones: Ver, Modificar, Adicionar, Eliminar, Buscar, Habilitar, Deshabilitar, Gestionar direcciones IPestudiosy Gestionar centros.

## Anexos

---

3. Selecciona la opción "Gestionar centros".	4. Muestra un listado de los centros existentes en el sistema y las opciones: Ver, Modificar, Adicionar, Buscar, Eliminar, Habilitar, Deshabilitar y Gestionar direcciones IP centros.
5. Selecciona la acción Habilitar centro.	6. Activa el centro especificado y a su vez todas las funcionalidades asociadas al mismo.
	7. El caso de uso termina.

Tabla 20. Descripción del caso de uso: *Habilitar centro*.



## ANEXO 2

### Diagramas de clases del diseño

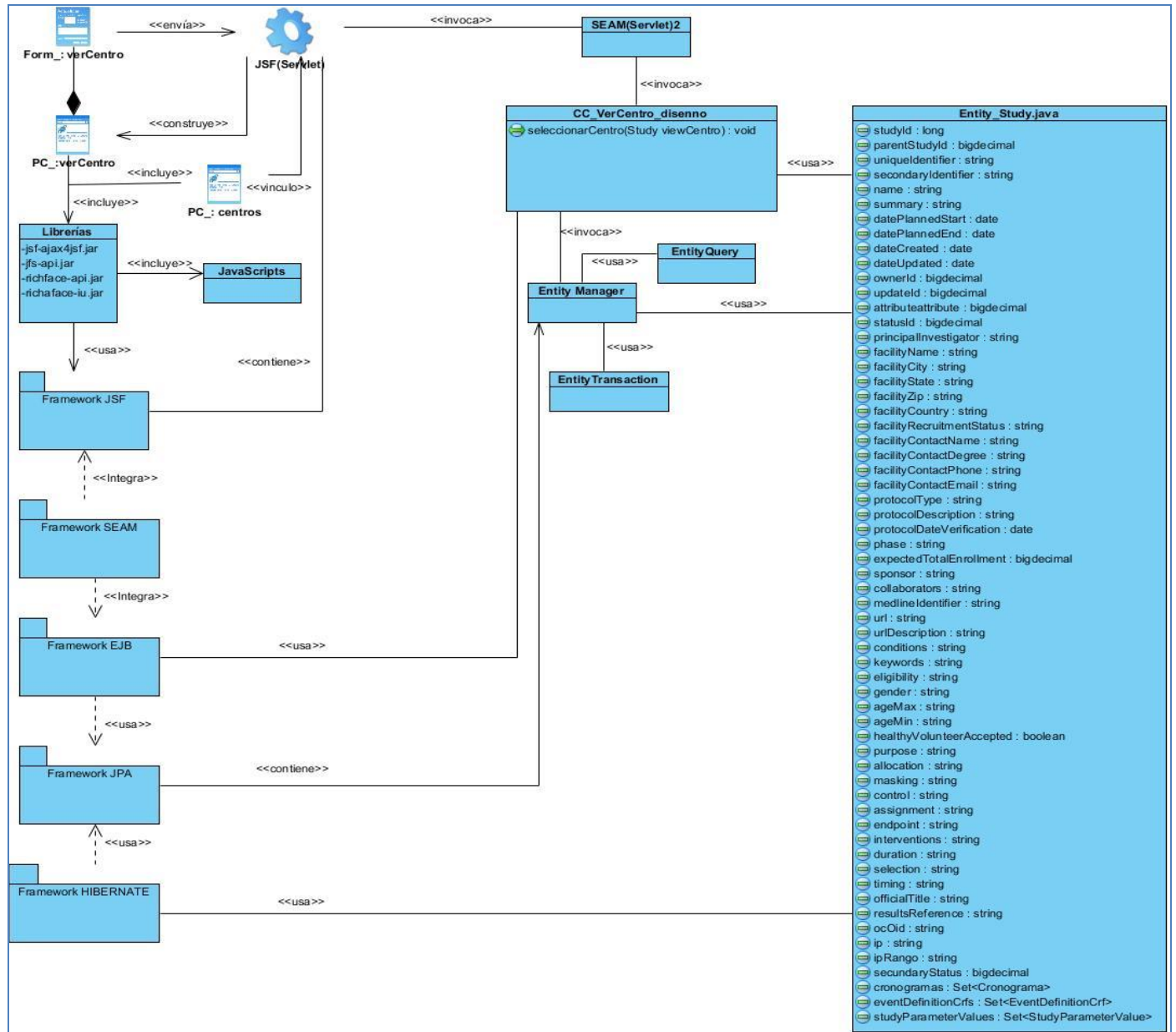


Fig. 12. Diagrama de clases del diseño: *Ver detalles de centro*

# Anexos

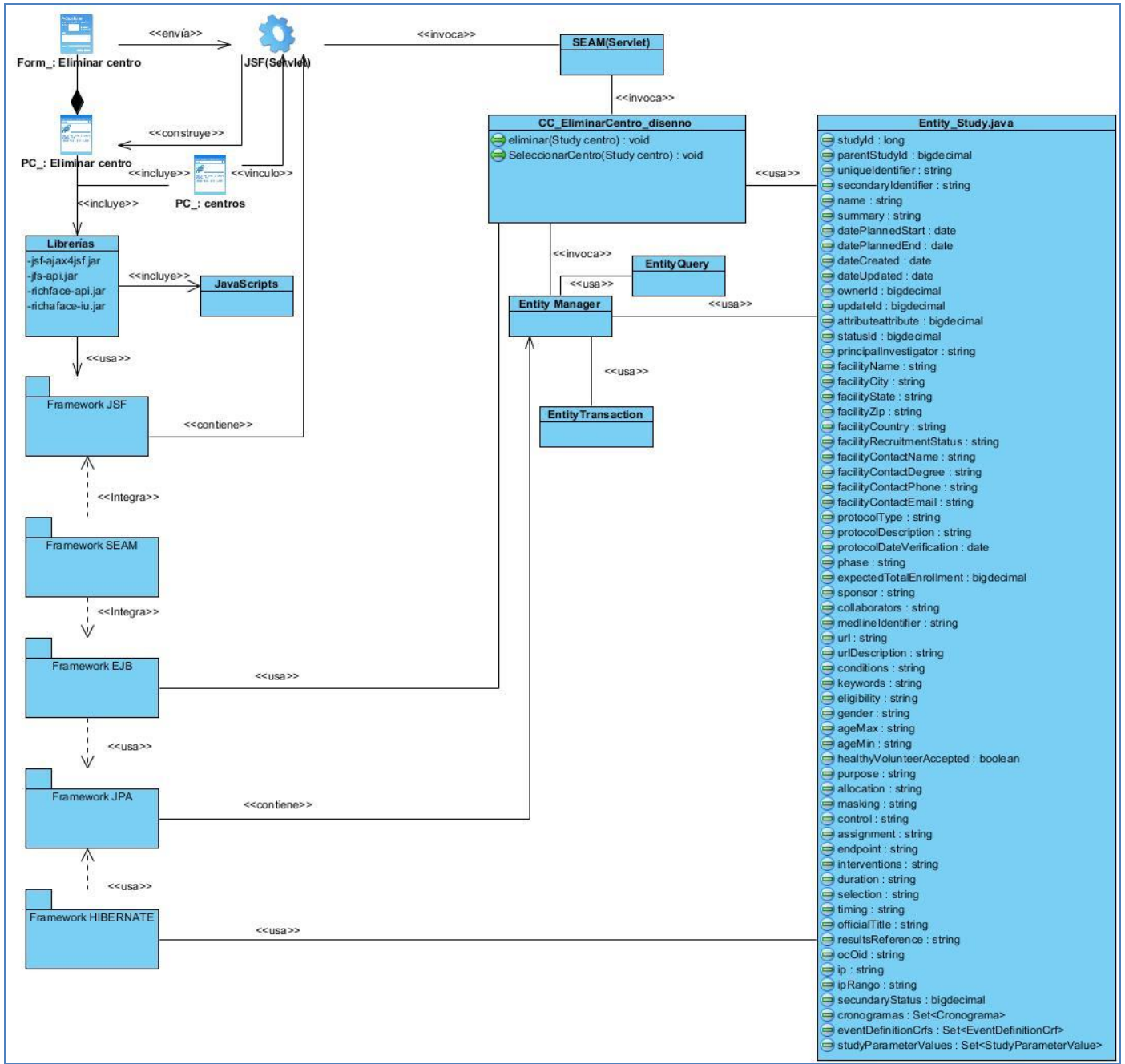


Fig. 13. Diagrama de clases del diseño: *Eliminar centro*

# Anexos

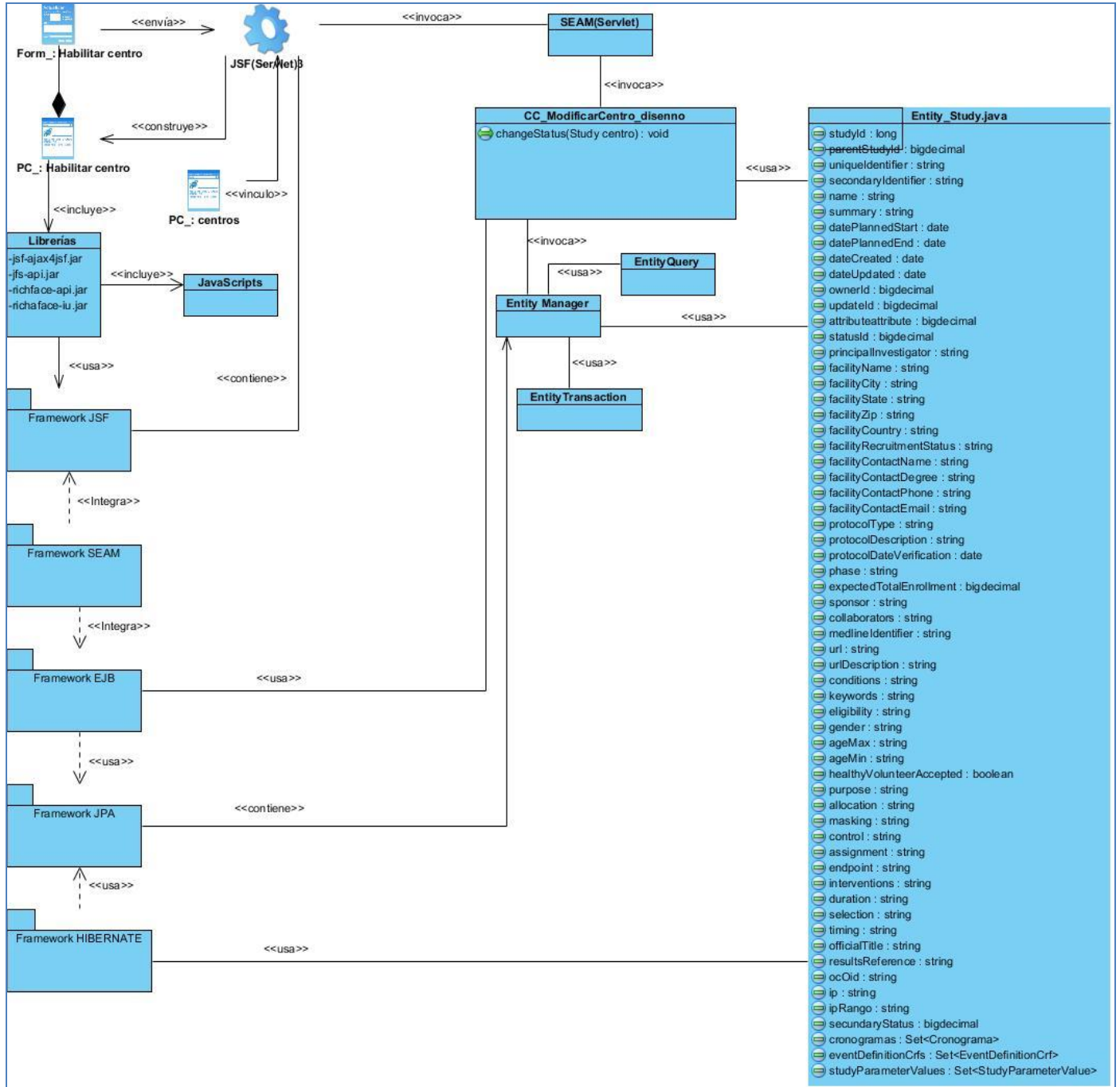


Fig. 14. Diagrama de clases del diseño: *Habilitar centro*